Alma Mater Studiorum · Università di Bologna

Campus di Cesena
Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea in
Ingegneria e Scienze Informatiche

CONNECTED VEHICLES: FROM CAN BUS TO IP-BASED IVN

Tesi di Laurea in
Reti di Telecomunicazione

Relatore:
Prof. Ing.
FRANCO CALLEGATI

Candidato:
JUAN SEBASTIAM SANCHEZ

Anno Accademico 2018/2019
Sessione IV

Ai miei genitori,
per avermi fatto diventare ciò che sono
ed avermi sostenuto in ogni momento.

Alla mia famiglia,
ai miei amici
e alle persone più care,
per avermi sempre incoraggiato.

A UniBo Motorsport,
per questi anni indimenticabili.

A tutti voi,
grazie.

Indice

Capitolo 1

Introduzione

Il settore automotive, negli ultimi vent'anni, è stato oggetto di importanti sviluppi tecnologici, caratterizzati principalmente dall'evoluzione dei settori dell'elettronica e delle telecomunicazioni.

Questo elaborato si pone come obiettivo lo studio delle tecnologie che hanno permesso l'introduzione di sistemi elettronici avanzati all'interno dei veicoli, e di come queste si siano evolute negli anni.

La tesi inizia con un'analisi dettagliata dello stato dell'arte della connettività veicolare. La materia è estremamente vasta, ma viste le finalità specifiche di questo studio si è scelto di introdurre solo i concetti essenziali per una comprensione completa delle problematiche e delle soluzioni proposte. Viene quindi presentata in primo luogo la moderna idea di *Connected Vehicle*, nonché tutte le proprietà che caratterizzano questo tipo di veicoli. Si procede poi introducendo quello che sarà invece il tema principale dell'elaborato: le *In-Vehicle Networks* ed i principali protocolli di comunicazione che ne hanno caratterizzato l'evoluzione.

Nel terzo capitolo viene dettagliatamente analizzato il *Controller Area Network*, o *CAN bus*, che rappresenta ad oggi lo standard di comunicazione seriale maggiormente utilizzato per il dialogo tra dispositivi onboard.

Il quarto capitolo si pone l'obiettivo di presentare in modo quanto più chiaro e completo possibile le *IP-based In-Vehicle Networks*, la tecnologia che si prevede dominerà il mondo delle comunicazioni veicolari.

Nel quinto capitolo viene affrontato un tema di primaria importanza per quanto riguarda l'implementazione di reti onboard eterogenee (o *multiprotocollo*), l'*Automotive Gateway*, di cui si analizzano le principali caratteristiche ed un framework opensource che ne permette una semplice implementazione.

Il documento si conclude infine con una breve analisi di tutti gli aspetti presi in esame ed una considerazione personale su ciò che caratterizzerà l'evoluzione futura di questi sistemi.

Capitolo 2

Stato dell'Arte

In questo capitolo viene affrontato un approfondimento generale sull'idea moderna di *automobile connessa* e tutte le principali tecnologie che ne permettono l'implementazione, sia in termini globali (connessione ad internet) che locali (reti onboard).

In particolar modo viene analizzato il concetto di *"Connected Vehicle"* [1], la sua evoluzione negli ultimi vent'anni, le principali infrastrutture e tecnologie che ne permettono il continuo sviluppo ed il modello topologico di base per un veicolo di questo genere. Saranno quindi analizzate le principali modalità di connessione e le differenti categorie di applicazioni attualmente utilizzate nell'integrazione tra la vettura e l'ambiente circostante.

Verranno successivamente introdotte le *In-Vehicle Networks* (IVN) [2] [3], andando ad analizzare i principali protocolli finora utilizzati. Sarà infine presentato uno step chiave nell'evoluzione di questi sistemi, le IP-based IVN.

2.1 Connected Vehicles

Con la terminologia *"Connected Vehicles"* [1] si vuole raggruppare l'insieme di vetture, di nuova concezione, capaci di comunicare in maniera bidirezionale con altri sistemi. In questo modo si permette al veicolo di condividere l'accesso ad internet con altri dispositivi, interni o esterni ad esso che siano, e di creare flussi di dati tanto in uscita quanto in entrata dalla vettura. Nasce così una nuova visione dell'automobile, in grado di agevolare la vita quotidiana dei guidatori. Non è più il conducente a dover cercare informazioni, ma è il sistema stesso a presentare una vista sempre aggiornata di tutto l'ambiente circostante, attraverso, ad esempio, notifiche automatiche riguardanti incidenti, percorsi da evitare, eccessi di velocità e tante altre informazioni provenienti da un bene di dominio pubblico, internet.

2.1.1 Storia: 1996 - oggi

General Motors è stata la prima casa automobilistica a mettere sul mercato, nel 1996, una vettura che presentava i primi cenni di connettività. Lo scopo principale del progetto era la sicurezza dei passeggeri: in caso di problematiche era possibile avviare una telefonata indirizzata ad un call center, dal quale veniva fornita assistenza immediata ai veicoli incidentati. Inizialmente il sistema funzionava esclusivamente con la voce, ma con l'avvento degli smartphone e del traffico dati è stato possibile integrare l'invio automatizzato di informazioni quali le coordinate GPS dell'incidente o del mezzo in avaria.

L'avvento del nuovo millennio portò grandi innovazioni nel settore automotive, prima tra tutte la diagnostica remota. Già a partire dal 2003 i servizi di automobili connesse includevano rapporti periodici alle case produttrici sulla salute dei veicoli, navigazione GPS passo-passo e dispositivi di accesso alla rete.

Nell'estate del 2014 Audi introduce un'incredibile novità: vengono offerti Hotspot Wi-Fi 4G LTE sui principali modelli della casa. Seguì a breve la prima distribuzione di massa di tali optional, diretta da General Motors, che in poco più di un anno aveva già processato oltre un miliardo di richieste dai clienti.

Nel 2017 la start-up Stratio Automotive munisce oltre 10.000 veicoli di intelligenza predittiva, fornendo alle grandi case automobilistiche un'innovativa gestione della manutenzione delle proprie vetture. A partire da questo momento infatti, sarà il sistema onboard ad indicare in totale autonomia al conducente, ed al tempo stesso alla casa produttrice, eventuali guasti o criticità di qualsiasi componente dell'automobile.

2.1.2 Topologia

Con l'evoluzione della componentistica di bordo delle vetture moderne e dei sistemi in esse implementati, diventa fondamentale avere a disposizione vere e proprie reti on board che permettano il dialogo tra le differenti piattaforme e che facilitino l'integrazione dei dispositivi per la connessione del veicolo ad internet.

Evidente come venga quindi abbandonato il concetto di cablaggio come "semplice" insieme di connessioni seriali tra le parti, dando spazio a quelle che sono ormai diventate complesse reti in cui sono presenti molteplici canali di comunicazione diversi, ognuno dedicato ad uno scambio di dati ben definito attraverso un protocollo ottimizzato per quella specifica applicazione. Diventa quindi essenziale anche l'introduzione di un dispositivo in grado di integrare le diverse subnet ed unificare in un solo protocollo (generalmente IP) la comunicazione con l'esterno, il *gateway*.

In Figura 2.1 è riportato un esempio basilare della topologia bus comunemente implementata nelle moderne vetture di serie:

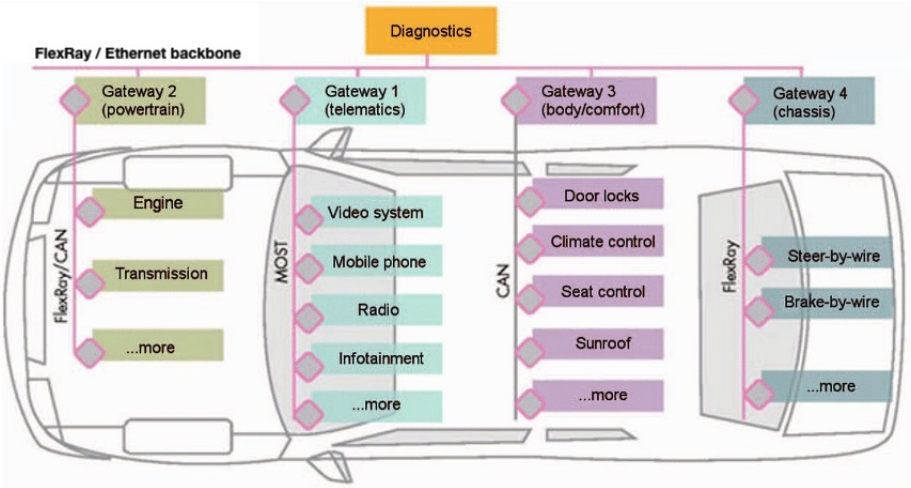

Figura 2.1: Esempio di moderna rete veicolare.

2.1.3 Tipi di connessione

Esistono cinque modi in cui un veicolo può essere collegato all'ambiente circostante e comunicare con esso:

Vehicle to Infrastructure (V2I)

La tecnologia cattura i dati generati dalla vettura e fornisce informazioni sull'infrastruttura al conducente. Vengono condivise informazioni su sicurezza, mobilità e condizioni ambientali.

Vehicle to Vehicle (V2V)

La tecnologia comunica informazioni su velocità e posizione dei veicoli circostanti attraverso uno scambio di informazioni wireless. L'obiettivo è quello di evitare incidenti, alleviare le congestioni del traffico e limitare l'impatto ambientale.

Vehicle to Cloud (V2C)

La tecnologia scambia informazioni su e per le applicazioni della vettura con un sistema cloud. Ciò consente al veicolo di utilizzare informazioni esterne, ottenute grazie all'integrazione tramite cloud di industrie, trasporti ed abitazioni intelligenti.

Vehicle to Pedestrian (V2P)

La tecnologia rileva informazioni sull'ambiente circostante e le comunica ad altri veicoli, infrastrutture e dispositivi mobili personali. Ciò consente alle vetture di comunicare con pedoni, migliorando così sicurezza e mobilità su strada.

Vehicle to Everything (V2X)

La tecnologia interconnette tutti i tipi di veicoli e infrastrutture, compresi veicoli stradali, navi, treni, aerei ed autostrade.

2.1.4 Mezzi di connessione ad Internet

Tra le principali barriere che hanno rallentato lo sviluppo dei veicoli connessi vi è la modalità di accesso ad internet. I clienti, infatti, sono spesso restii a dover sostenere spese aggiuntive associate alla connettività integrata, appoggiando invece le soluzioni alternative che prevedono l'utilizzo del proprio smartphone, e del relativo piano dati, come link tra la vettura ed internet. Ne segue, quindi, un impegno sempre maggiore delle case automobilistiche nell'integrazione dei dispositivi mobili per soddisfare la richiesta di connettività da parte dei clienti.

Come anticipato, sono due le differenti possibilità per l'accesso al web:

Brought-in devices

Dispositivi collegati alla porta *On-Board Diagnostics* (OBD) [4] attraverso la quale ricevono alimentazione ed accesso ai dati veicolo. Possono essere a loro volta suddivisi per tipologia di connessione ad internet:

- Connessione tramite smartphone del guidatore
- Connessione tramite modulo GSM dedicato

Built-in telematics boxes

Dispositivi integrati nel sistema IT del veicolo, che nella maggior parte dei casi sono provvisti di connessione ad internet propria attraverso un modulo GSM.

2.1.5 Categorie di applicazioni

Grazie al forte sviluppo delle infrastrutture e dei sistemi di bordo delle vetture moderne, nonché della loro connettività, è stato possibile/necessario implementare applicazioni di bordo che migliorassero l'esperienza di guida dei passeggeri e la sicurezza del traffico urbano. Ne segue quindi una divisione in due classi principali:

Single Vehicle Applications

Contenuti necessari ad arricchire il sistema di infotainment ed applicazioni di servizio implementate dalla vettura in autonomia, in connessione con un servizio cloud o backoffice. Non vi è quindi comunicazione con altri veicoli o infrastrutture per lo scambio di dati necessari al funzionamento delle applicazioni.

Fanno parte di questa categoria le applicazioni volte, ad esempio, alla ricerca di parcheggi o stazioni di servizio, nonché la gestione degli avvisi al conducente per l'arrivo in orario ad appuntamenti segnati sul calendario e i sistemi di streaming audio on demand.

Cooperative Safety and Efficiency Applications

Prevedono la comunicazione tra veicoli e/o infrastrutture. Attraverso una forte collaborazione tra le parti, permettono l'implementazione di features di estrema importanza per il miglioramento della sicurezza e della mobilità.

Questi sistemi sono principalmente legati alle moderne tecnologie ADAS (*Advanced Driver-Assistance Systems*) [5], che sulla base di input provenienti anche dai veicoli circostanti, riescono ad attuare reazioni istantanee basate su attività automatiche di monitoraggio, allerta, frenata e sterzata.

Ovviamente, in questo contesto il principale ostacolo ad implementazioni avanzate è il quadro normativo nazionale ed internazionale. Per raggiungere il completamento si sistemi efficienti ed efficaci devono essere affrontate questioni come privacy e sicurezza, portando dunque all'inevitabile necessità di regolamentazioni ben strutturate e condivise tra le parti. La US National Highway Traffic Safety Administration (NHTSA) ha sostenuto la regolamentazione della comunicazione V2V, avviando il processo di elaborazione delle regole nel dicembre 2016. La proposta prevede l'introduzione di comunicazioni a corto raggio (DSRC [6]) nei veicoli leggeri, che saranno quindi obbligati a trasmettere in modo broadcast un pacchetto predefinito, il *Basic Safety Message* (BSM [7]), fino a dieci volte al secondo, indicando posizione, direzione e velocità del veicolo. Nell'Unione Europea, invece, non vi è ancora nessun movimento verso l'obbligazione dei produttori all'introduzione di sistemi normati per la connettività.

2.1.6 Intelligent Transportation System (ITS)

Un *sistema di trasporto intelligente* (in inglese ITS) [8] è un'applicazione avanzata che mira a fornire servizi innovativi relativi alle diverse modalità di trasporto e gestione del traffico, per consentire agli utenti di essere meglio informati ed utilizzare così le reti di trasporto in modo più sicuro, più coordinato e più intelligente. Alcune di queste tecnologie includono la richiesta di servizi di emergenza in caso di incidente,

l'uso di telecamere per far rispettare le leggi sul traffico o segnali che contrassegnano le variazioni del limite di velocità in base alle condizioni di un'intera rete stradale.

Come intuibile, questi sistemi non richiedono la collaborazione tra più parti per ottenere il risultato voluto, ma la semplice comunicazione da parte dell'infrastruttura (in genere) di informazioni utili al guidatore.

2.1.7 Cooperative Vehicle Infrastructure System (CVIS)

Nati come sviluppo dei più semplici ITS, i *sistemi cooperativi veicolo-infrastruttura* (CVIS) [9] possono acquisire informazioni su mezzi di trasporto e strade utilizzando tecnologie di comunicazione wireless e di rilevamento dei sensori, consentendo l'interazione e la condivisione dei dati tra veicoli, o tra veicoli e infrastrutture. Tale sistema è una buona soluzione per comunicazione e coordinamento intelligenti V2I, rendendo l'utilizzo delle risorse del sistema più efficiente e consentendo un traffico stradale più sicuro e meno caotico.

Un CVIS [9] implementa un'interazione che interpreta con grande precisione le intenzioni dei partecipanti al traffico. Non solo indovina cosa farà un'auto, ma cattura la situazione in modo estremamente accurato, creando un contesto informativo sulla base del quale il sistema sarà in grado di fare valutazioni e prendere decisioni in tempi estremamente minori rispetto ai normali tempi di reazione dell'essere umano.

Un'infrastruttura avanzata di questo genere è quindi in grado di fornire sufficienti istruzioni per il processo decisionale dei veicoli autonomi, i quali saranno così soggetti ad una notevole riduzione di complessità e costi, non essendo più necessaria la preventiva elaborazione di tutti i possibili scenari. La guida autonoma vede così una possibilità di effettiva commercializzazione a corto raggio, implementabile in un futuro non così lontano.

2.2 In-Vehicle Networks (IVN)

Con il termine *In-Vehicle Network* [3] si vuole identificare una rete di comunicazione specializzata che collega i diversi moduli all'interno di un veicolo, di qualunque tipo esso sia (automobile, autobus, treno, mezzo industriale o agricolo, nave o aereo).

Un modulo di controllo elettronico in genere riceve da sensori degli input periodici, da cui, attraverso una serie predefinita di calcoli, deriva informazioni e comandi da poi imporre a degli attuatori (accendere la ventola di raffreddamento, cambiare marcia, ecc.). I differenti dispositivi presenti a bordo di un comune mezzo di trasporto devono quindi poter scambiare dati tra loro in modo rapido ed affidabile, caratteristiche che hanno portato al necessario sviluppo delle IVN come mezzo avanzato di scambio di dati.

Ovviamente, per rendere quanto più sicuro ed efficace il controllo dei veicoli, viene richiesto di soddisfare requisiti speciali, diversi da quanto richiesto nelle più comuni reti telematiche. La garanzia della consegna dei messaggi e del rispetto dei tempi massimi per recapitarli, l'inesistenza quasi totale di conflitti tra i pacchetti, il routing ridondante e la riduzione massimale dei costi sono solo alcune delle caratteristiche che vincolano in questo caso all'uso di protocolli di rete meno comuni.

2.2.1 Protocolli

Esistono diversi tipi di rete e protocolli utilizzati nei veicoli dai vari produttori. Molte case automobilistiche hanno incoraggiato, e continuano a farlo, la standardizzazione di un solo protocollo di comunicazione, che ancora non si è raggiunta per molteplici motivi. Allo stato attuale, sono quattro i principali protocolli che si usa implementare in maniera parallela su quasi ogni mezzo di trasporto, ognuno dei quali viene sfruttato per i propri vantaggi.

Controller Area Network (CAN bus)

Robusto standard di comunicazione progettato principalmente per applicazioni automotive, si basa sullo scambio di messaggi tra i dispositivi della rete, ognuno dei quali invia i dati di un frame in maniera sequenziale. Se più di un dispositivo trasmette contemporaneamente, allora il dispositivo con la priorità più alta è in grado di continuare la trasmissione, mentre tutti gli altri si mettono in ascolto.

Caratterizzato dalla sua robustezza, semplicità implementativa e costo contenuto, rappresenta ad oggi il protocollo più diffuso sui comuni mezzi di trasporto e per questo motivo sarà largamente analizzato nel successivo capitolo.

Local Interconnect System (LIN)

LIN [10] è un protocollo di comunicazione seriale economico, particolarmente indicato per nodi meccatronici in applicazioni automobilistiche distribuite, ma ugualmente adatto per applicazioni industriali.

Gli usi attuali ne combinano l'efficienza a basso costo con semplici sensori per creare piccole sottoreti che integrano la rete CAN [11] principale del veicolo, permettendo così la creazione di semplici reti gerarchiche all'interno dell'automobile.

Tutti i messaggi vengono inviati dal master, con al massimo uno slave che risponde ad un determinato identificatore del messaggio, perciò non risulta necessario implementare un sistema di rilevamento delle collisioni.

Tra le principali caratteristiche di questo protocollo troviamo:

- Tempi di latenza garantiti
- Lunghezza variabile del frame di dati (2, 4 e 8 byte)
- Flessibilità di configurazione
- Checksum dei dati e rilevamento degli errori
- Rilevamento di nodi difettosi

Viene comunemente utilizzato anche per la faciilità d'uso e di implementazione delle estensioni, la convenienza (in termini monetari) rispetto al CAN bus [11] ed altri standard di comunicazione, nonché l'assenza di costi di licenza.

LIN [10] non si pone quindi in alcun modo come alternativa al CAN bus [11], ma rimane una buona soluzione laddove sia essenziale mantenere costi contenuti e né velocità né larghezza di banda sono un fattore prioritario. In genere, viene utilizzato all'interno di sottosistemi che non sono fondamentali per le prestazioni o la sicurezza del veicolo, come il controllo del posizionamento dei sedili, del climatizzatore o dell'illuminazione.

FlexRay

FlexRay [12] è un protocollo di comunicazione per reti automobilistiche, sviluppato dal consorzio *FlexRay* per governare il computing automobilistico di bordo. È progettato per essere più veloce e più affidabile del CAN bus [11], ma è anche più costoso, motivo per il quale il progetto è stato abbandonato nel 2009. Tuttavia, le specifiche sono ancora disponibili ed il protocollo è stato convertito in standard.

Supporta velocità di trasferimento dati fino a 10 Mbit/s, topologie a stella e può avere due canali dati indipendenti per la tolleranza agli errori (la comunicazione può continuare con una larghezza di banda ridotta se un canale non è operativo).

Il bus funziona su un ciclo temporale, diviso in due parti: il *Segmento Statico* e il *Segmento Dinamico*. Il *Segmento Statico* è preallocato in sezioni per i singoli tipi di comunicazione, fornendo un determinismo più forte rispetto al suo predecessore CAN. Il *Segmento Dinamico* funziona più come il CAN bus [11], con i nodi che assumono il controllo del bus quando disponibile, consentendo un comportamento innescato da eventi.

Le principali applicazioni che vedono FlexRay ancora come protocollo dominante sono le tecnologie ADAS [5], *X-by-wire* [13] (Throttle-by-wire, Brake-by-wire, ecc.) e tutte quelle in cui sono richiesti elevati standard prestazionali.

Il bus presenta alcuni svantaggi come livelli di alimentazione inferiori a quelli utilizzati per tutti gli altri standard e asimmetria dei bordi, il che comporta problemi nell'estensione della rete. Per questo morivo si prevede che Ethernet sostituirà FlexRay [12] per le applicazioni che richiedono molta larghezza di banda.

IP-based IVN e Automotive Ethernet

Nonostante Ethernet sia uno standard da oltre 20 anni, non ha trovato importanti applicazioni nel settore automotive fino a qualche anno fa a causa di quattro principali limiti:

- Non soddisfava i requisiti OEM EMI/RFI per il mercato automobilistico: tali reti erano infatti molto suscettibili alle interferenze elettromagnetiche, spesso create dai dispositivi elettronici presenti in una vettura;
- Non era in grado di garantire la latenza sotto il microsecondo: fattore chiave per sostituire la comunicazione con qualsiasi sensore/controllo che necessitasse di tempi di reazione estremamente contenuti;
- Non permetteva di controllare l'allocazione della banda a flussi diversi, quindi non poteva essere usato per trasmettere dati condivisi da fonti di diverso tipo;
- Non forniva la possibilità di sincronizzazione del tempo tra i dispositivi.

Automotive Ethernet [14] si presenta quindi come una versione ottimizzata per l'uso veicolare del protocollo da cui prende il nome. Per soddisfare pienamente tali requisiti, molteplici nuove specifiche e revisioni di quelle esistenti sono state eseguite dai gruppi IEEE 802.3 e 802.1.

In passato utilizzato principalmente per sistemi diagnostici, di infotainment ed il collegamento di sensori remoti, trova ora invece sempre più piede in tutti i sistemi che richiedono una maggiore larghezza di banda per trasmettere i dati alle velocità necessarie per mantenere la sicurezza del conducente, velocità che reti come CAN e FlexRay non sono in grado di fornire. Inoltre, grazie alla sua scalabilità permette l'introduzione di switch attraverso i quali diventa possibile connettere alla rete un numero indefinito di nodi, riducendo la quantità di cablaggio richiesta e quindi il peso complessivo della vettura.

Se si considerano i vantaggi in termini di costi e di compatibilità, quando si collegano i veicoli ad un'infrastruttura intelligente, è logico che Ethernet guidi la carica nella futura connettività V2X. CAN [11], CAN-FD e LIN [10] rimarranno probabilmente rilevanti per il medio termine: sono affermati, convenienti e rimarranno convenienti per le soluzioni in cui basso costo e larghezza di banda ridotta sono specifiche chiave di progettazione. È però quasi certo che a lungo termine Automotive Ethernet [14] arriverà a sostituire le classiche IVN [3], creando tuttavia nuove sfide nella sicurezza informatica per garantire la compatibilità con l'infrastruttura già esistente.

In figura 2.2 si mostra l'evoluzione temporale di Automotive Ethernet [14] avvenuta negli scorsi anni:

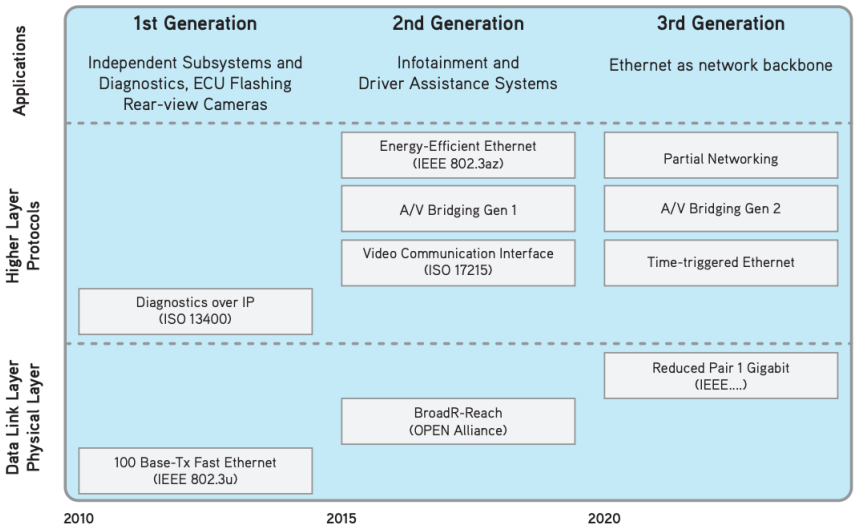

Figura 2.2: Sviluppo di Automotive Ethernet nell'ultimi decennio.

Capitolo 3

Controller Area Network

Nel seguente capitolo si affronta un'analisi dettagliata del *Controller Area Network* (CAN) [15], che come già anticipato, si presenta ad oggi come lo standard di comunicazione più diffuso in ambito veicolare ed industriale.

Viene quindi proposta un'overview generale del protocollo, che ne introduce la storia, i principali benefici, le applicazioni di maggior rilievo, il modello di comunicazione e le altre principali caratteristiche.

Continuando, vengono poi analizzate la pila protocollare, la struttura dei messaggi, le varie tecniche di gestione degli errori e i principali protocolli standardizzati di alto e basso livello.

3.1 Overview

Il *Controller Area Network* [15], noto anche come *CAN-bus*, è uno standard seriale per bus di campo (pensato principalmente per l'ambiente automotive) di tipo multicast, introdotto negli anni ottanta per collegare unità di controllo elettronico (ECU). Il CAN è stato espressamente progettato per funzionare senza problemi in ambienti fortemente disturbati dalla presenza di onde elettromagnetiche e può utilizzare come mezzo trasmissivo una linea a differenza di potenziale bilanciata, come la RS-485 [18]. L'immunità ai *disturbi elettromagnetici* [16] può essere ulteriormente aumentata utilizzando cavi a doppino intrecciato.

3.1.1 Storia

Lo sviluppo del CAN bus iniziò nel 1983 presso la Robert Bosch GmbH [19], ma il protocollo è stato ufficialmente rilasciato solo nel 1986 alla conferenza della *Society of Automotive Engineers* (SAE) [20] a Detroit. Rilasciato nel 1991, il

Mercedes-Benz W140 fu il primo veicolo di produzione a disporre di un sistema di cablaggio multiplex basato su CAN.

Bosch nel tempo ha pubblicato diverse versioni della specifica CAN, l'ultima delle quali è stata la CAN 2.0, pubblicata nel 1991. Questa specifica ha due parti: la *parte A* è per il formato standard (con identificatore a 11 bit), mentre la *parte B* è per il formato esteso (con identificatore a 29 bit).

Nel 1993, l'*International Organization for Standardization* (ISO) [21] ha rilasciato lo standard CAN ISO 11898, che è stato successivamente ristrutturato in due parti: ISO 11898-1 [32], che definisce le specifiche del livello data link, e ISO 11898-2 [33], che si occupa dello strato fisico del CAN ad alta velocità. La ISO 11898-3 [34], rilasciata in un secondo momento, si occupa dello strato fisico del low-speed/fault-tolerant CAN. Gli standard ISO 11898-2 e ISO 11898-3 non fanno parte delle specifiche Bosch CAN 2.0, ma possono essere acquistati dall'ISO.

Bosch è ancora attiva nell'estensione degli standard CAN: nel 2012 ha infatti rilasciato CAN FD 1.0, o Flexible Data-Rate CAN [39]. Questa specifica utilizza un formato di frame diverso e consente una diversa lunghezza dei dati, nonché il passaggio ad un bit-rate più elevato. CAN FD è compatibile con le reti 2.0 esistenti, quindi i nuovi dispositivi possono coesistere sulla stessa rete con i dispositivi esistenti.

3.1.2 Benefici

Grazie ai notevoli progressi tecnologici, ad oggi il protocollo vanta notevoli vantaggi in termini di:

- *Leggerezza e basso costo*: CAN fornisce una rete economica e duratura che consente a più dispositivi di comunicare tra loro. Le centraline elettroniche (ECU [17]) possono avere una singola interfaccia CAN anziché molteplici input analogici e digitali per ogni dispositivo nel sistema, riducendo così il costo ed il peso complessivi delle automobili;

- *Comunicazione broadcast*: ogni dispositivo sulla rete legge tutto il traffico di dati e in maniera indipendente decide se un messaggio è rilevante o se deve essere filtrato. Questa struttura consente modifiche alle reti CAN con un impatto minimo: nuovi nodi non trasmittenti possono essere aggiunti senza modifiche di nessun tipo alla rete;

- *Priorità*: ogni messaggio ha una priorità, quindi se due nodi tentano di comunicare contemporaneamente, quello con priorità maggiore viene trasmesso e l'altro viene posticipato. Questo arbitrato non è distruttivo, ma si traduce,

se necessario, ad una trasmissione ininterrotta del messaggio con la massima priorità, consentendo così alle reti di soddisfare i vincoli di tempistica deterministica;

- *Gestione degli errori*: la specifica CAN include un codice di ridondanza ciclica (CRC) per eseguire il controllo degli errori sul contenuto di ciascun frame. I messaggi con errori vengono ignorati da tutti i nodi ed un frame di errore può essere trasmesso per segnalare il problema alla rete. Gli errori globali e locali vengono differenziati dai singoli dispositivi e, nel caso in cui siano rilevati troppi errori, i singoli nodi possono disconnettersi completamente dalla rete in maniera autonoma.

3.1.3 Applicazioni

CAN è stato creato per uso automobilistico, quindi la sua applicazione più comune è la *rete elettronica di bordo* (IVN [3]). Tuttavia, poiché altri settori hanno compreso l'affidabilità e i vantaggi di questo protocollo, negli ultimi 20 anni è stato adottato da un'ampia varietà di settori diversi. Le applicazioni ferroviarie come tram, metropolitane, ferrovie leggere e treni a lunga percorrenza implementano il CAN bus, così come viene adottato anche su aeromobili con sensori di stato di volo, sistemi di navigazione e altri moduli avanzati.

I produttori di apparecchiature mediche utilizzano CAN come rete integrata nei loro dispositivi, mentre alcuni ospedali utilizzano il protocollo addirittura per la gestione di sale operatorie da remoto.

La specifica *CANopen* [22] del protocollo è utilizzata anche in applicazioni non industriali come attrezzature da laboratorio, telecamere sportive, telescopi, porte automatiche e persino macchine da caffè.

3.1.4 Struttura fisica

I nodi di una rete CAN sono collegati tra loro tramite un bus a due fili, comunemente conosciuto come Twisted Pair (doppino), ovvero una coppia intrecciata di conduttori con un'impedenza caratteristica di 120 ohm (nominale). Il bus utilizza due segnali differenziali, CAN *alto* (CANH) e CAN *basso* (CANL), che possono essere portati ad uno stato *dominante* (con CANH maggiore di CANL), oppure trascinati da resistori passivi (chiamati *terminazioni*) verso uno stato *recessivo* (con CANH minore o uguale a CANL).

In Figura 3.1 viene riportata la composizione generica di una rete CAN:

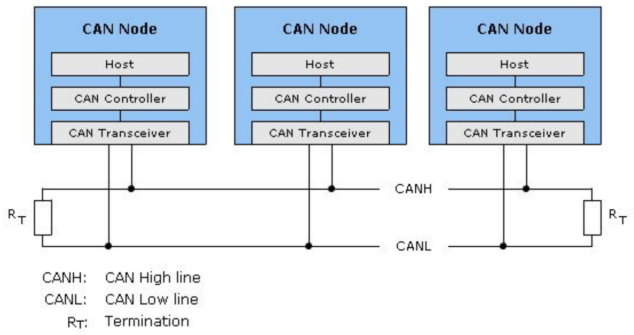

Figura 3.1: Composizione generica di una rete CAN.

La specifica CAN ad alta velocità guida il segnale CANH verso 5V e CANL verso 0V quando un dispositivo trasmette un bit dominante. In caso contrario, i resistori di terminazione riportano i due segnali al livello recessivo, con una tensione differenziale nominale di 0V (i ricevitori considerano recessiva qualsiasi tensione differenziale inferiore a 0.5V). La tensione differenziale *dominante* equivale a 2V.

Lo standard a bassa velocità e fault-tolerant funziona in modo simile al protocollo ad alta velocità, ma con oscillazioni di tensione maggiori. Lo stato dominante viene trasmesso portando CANH verso la tensione di alimentazione del dispositivo (5V o 3.3V) e CANL verso 0V quando viene trasmesso un bit dominante, mentre le resistenze di terminazione portano il bus in uno stato recessivo con CANH a 0V e CANL a 5V. Ciò consente un ricevitore più semplice che considera solo il segno di $CANH - CANL$. Per un corretto funzionamento, il mezzo trasmissivo deve essere in grado di gestire da -27V a $+40$V senza problemi.

3.1.5 Nodi della rete

Ogni nodo della rete deve essere composto da:

- *CPU*: determina il significato dei messaggi ricevuti e i frame che desidera trasmettere; ad esso possono essere collegati sensori, attuatori ed altri dispositivi di controllo;

- *Controller* (spesso parte integrante della CPU): durante la fase di ricezione memorizza i bit seriali ricevuti sul bus fino a quando non è disponibile un intero messaggio, che può quindi essere recuperato dal processore (in genere è il Controller CAN ad attivare un interrupt); in fase di trasmissione, invece, il processore invia i messaggi da trasmettere al Controller, che ritrasmette i bit in serie sul bus non appena questo è libero;

- *Transceiver*: in ricezione si occupa di convertire il flusso di dati dai livelli CANbus ai livelli utilizzati dal Controller CAN; in trasmissione, invece, converte il flusso di dati dal Controller ai livelli CANbus.

3.1.6 Trasmissione dati

La trasmissione dei dati CAN utilizza un metodo di arbitrato bit a bit senza perdita per la risoluzione della contesa. Tale metodo richiede che tutti i nodi della rete siano sincronizzati per campionare tutti i bit contemporaneamente, motivo per cui il protocollo viene chiamato da alcuni "sincrono". Sfortunatamente però, tale termine è impreciso poiché i dati vengono trasmessi senza un segnale di clock, ovvero in modo asincrono.

Il protocollo trasmette i dati secondo un modello basato su bit *dominanti* (0 logici) e *recessivi* (1 logici). Se un nodo trasmette un bit dominante e un altro un bit recessivo, allora il bit dominante *vince* fra i due (realizzando un confronto di *AND* logico). In Figura 3.2 è riportata la tabella di verità di tale comparazione:

Stato del bus quando due nodi trasmettono		
	dominante	**recessivo**
dominante	dominante	dominante
recessivo	dominante	recessivo

AND logico		
	0	**1**
0	0	0
1	0	1

Figura 3.2: Tabella stati logici del canale CAN.

Con questa tecnica, quando viene trasmesso un bit recessivo, e contemporaneamente un altro dispositivo trasmette un bit dominante si ha una collisione, e solo il bit dominante è visibile in rete (tutte le altre collisioni sono invisibili). In termini prettamente elettrici, un bit dominante è dato dalla generazione di una tensione fra i conduttori, mentre un bit recessivo è semplicemente ignorato (differenza di potenziale nulla). In questo modo, quindi, si i è sicuri che ogni volta che si impone una differenza di potenziale, tutta la rete la rileva, e quindi è consapevole che si tratta di un bit dominante.

Solitamente si applica lo schema *Carrier Sense Multiple Access/Bitwise Arbitration* (CSMA/BA [23]): se due o più dispositivi iniziano a trasmettere contemporaneamente, si applica un meccanismo di arbitrato basato sulla priorità per decidere a quale dispositivo permettere di proseguire la trasmissione. Durante la comunicazione, ogni nodo in trasmissione controlla lo stato del bus e confronta il bit ricevuto con quello trasmesso. Se un bit dominante è ricevuto mentre un bit recessivo è trasmesso, il nodo interrompe la trasmissione. L'arbitrato è eseguito durante la trasmissione del pacchetto dati di identificazione del nodo. I nodi che

iniziano contemporaneamente a trasmettere inviano un ID dominante (0 binario), che incomincia con il bit alto. Non appena il loro ID è rappresentato da un numero più grande (quindi a priorità minore) i nodi stessi inviano un bit 1 (recessivo) e aspettano la risposta di uno 0 (dominante), quindi interrompono la trasmissione. Al termine dell'invio degli ID, tutti i nodi sono tornati allo stato di *OFF*, e il messaggio con la priorità massima può liberamente transitare.

Come si vede dalla Figura 3.3, vi è uno stretto legame tra la velocità con cui i nodi comunicano e la lunghezza del bus:

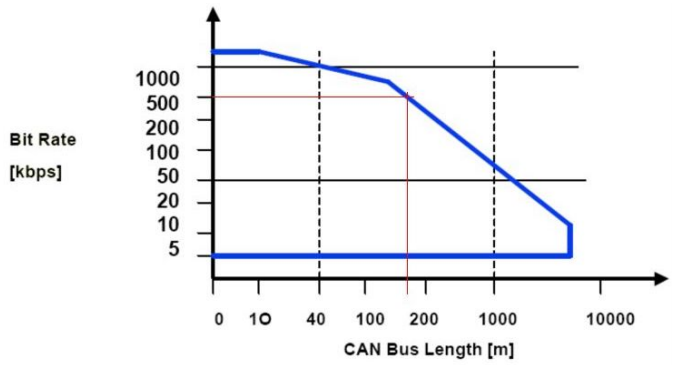

Figura 3.3: Rapporto tra data-rate e lunghezza della rete CAN.

È evidente come sia possibile raggiungere velocità fino a 1 Mbit/s a lunghezze di rete inferiori a 40 m, mentre come la riduzione della velocità consenta distanze di rete più lunghe (ad es. 500 m a 125 kbit/s circa).

3.1.7 Proprietà elettriche

Con il protocollo CAN, sia nelle implementazioni ad alta che a bassa velocità, il cambio di stato è più veloce quando si verifica una transizione da recessivo a dominante, poiché il mezzo trasmissivo viene guidato attivamente. In questo caso, la velocità esatta del cambio di stato dipende principalmente dalla lunghezza della rete e dalla capacità del mezzo utilizzato.

Il protocollo ad alta velocità viene normalmente utilizzato in applicazioni automobilistiche e industriali, dove bus attraversa l'*ambiente* (il veicolo o la macchina automatica) da un'estremità all'altra. Il low-speed/fault-toleran CAN viene invece utilizzata laddove gruppi di nodi devono essere collegati insieme.

Per funzionare correttamente, il bus CAN deve essere *terminato*. Le resistenze di terminazione sono necessarie per sopprimere i riflessi e riportare il bus allo stato recessivo, o inattivo. Il CAN ad alta velocità utilizza una resistenza da 120 ohm su ciascuna estremità di un bus lineare, mentre il protocollo a bassa velocità prevede l'utilizzo di resistori su ciascun nodo.

3.1.8 Sicurezza

CAN è un protocollo di basso livello che non supporta intrinsecamente alcuna funzionalità di sicurezza. Non esiste nessun sistema di crittografia in nessuna delle implementazioni standard, lasciando così queste reti aperte agli attacchi di tipo *Man-In-The-Middle* [24].

3.1.9 Licenze

Bosch detiene attualmente i brevetti sulla tecnologia, nonostante quelli relativi al protocollo originale siano ormai scaduti. I produttori di microprocessori compatibili con lo standard CAN pagano quindi le tasse di licenza a Bosch per l'uso del marchio CAN (e dei brevetti più recenti relativi a CAN FD [39]), che normalmente vengono trasferite al cliente finale nel prezzo del chip.

3.2 Livelli

Il protocollo CAN è strutturato su quattro diversi livelli gerarchici, in analogia con il modello ISO/OSI [25]. In figura 3.4 è riportato un esempio di tale gerarchia:

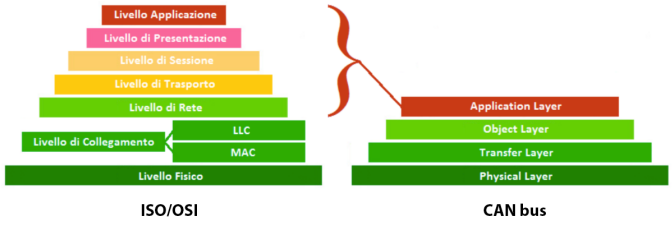

Figura 3.4: Pila protocollare dello standard CAN.

3.2.1 Application layer

Lo standard non prevede di per sé protocolli di livello applicativo volti ad interfacciare e fornire servizi ai processi in esecuzione sui singoli nodi. Ciò ha quindi richiesto l'implementazione di apposite specifiche, alcune standardizzate ed altre no, sviluppate direttamente dai produttori per soddisfare i propri requisiti di alto livello.

3.2.2 Object layer

Le proprietà dell'Object Layer dipendono strettamente dall'hardware specifico che lo implementa. Questo livello si occupa di attuare semplici ma fondamentali piani di gestione e filtraggio dei messaggi e del coordinamento dello stato del canale.

3.2.3 Transfer layer

La maggior parte dello standard CAN si applica a questo strato. Il livello, che riceve messaggi dal Physical layer e li trasmette all'Object layer, è responsabile della sincronizzazione dei bit, della definizione dei messaggi, dell'arbitrato e del rilevamento e segnalazione degli errori.

Vengono eseguite strategie per il controllo di:

- Rilevazione e confinamento errori
- Convalida, riconoscimento e contestualizzazione del messaggio
- Arbitrato
- Velocità di trasferimento e tempistiche
- Instradamento delle informazioni

3.2.4 Physical layer

Il protocollo inizialmente specificava solo il livello di collegamento, definendo in maniera astratta alcuni concetti per l'implementazione del livello fisico. Gli aspetti elettrici di questo strato (tensione, corrente, numero di conduttori) sono stati specificati successivamente nella norma ISO 11898-2 del 2003, mentre quelli meccanici (tipo e numero del connettore, colori, etichette, pin-out) devono ancora essere specificati formalmente. Tuttavia, sono emersi numerosi standard "de facto" per l'implementazione meccanica, il più comune dei quali è il connettore maschio D-sub a 9 pin.

L'immunità al rumore definita nell'ISO 11898-2 si ottiene mantenendo l'impedenza differenziale del bus ad un livello contenuto tramite resistori di basso valore (120 ohm) a ciascuna estremità del bus. Tuttavia, quando è inattivo, un bus a bassa impedenza come CAN assorbe più corrente (e potenza) rispetto ad altri protocolli basati sul controllo della tensione. Sui sistemi CAN, il funzionamento bilanciato del canale, in cui la corrente in una linea è esattamente bilanciata dalla corrente nella direzione opposta sull'altra linea, fornisce un riferimento di 0V indipendente e

stabile per tutti i nodi. Le migliori pratiche impongono che i segnali siano trasportati in doppini intrecciati di cavo schermato, per ridurre al minimo le emissioni RF e la suscettibilità alle interferenze nell'ambiente, già rumoroso, di un veicolo.

Durante uno stato recessivo (presente sul bus solo quando nessuno dei trasmettitori sta imponendo una condizione dominante), le linee di segnale e le resistenze rimangono in uno stato di alta impedenza rispetto ad entrambe i conduttori: CANH e CANL tendono quindi verso una tensione a metà tra le linee (di norma +2.5V circa). Durante uno stato dominante, invece, le linee di segnale e le resistenze si spostano in uno stato di bassa impedenza, in modo che la corrente fluisca attraverso i resistori. La tensione su CANH tende quindi a +5V, mentre su CANL tende a 0V.

3.3 Terminologia

Il protocollo CAN è caratterizzato dai tipi diversi di messaggi che possono essere scambiati tra i nodi della rete in base allo stato del canale. Inoltre, i vari sviluppi dello standard hanno portato all'introduzione di particolari strategie volte alla diminuzione/gestione degli errori.

3.3.1 Messaggi

Una rete CAN può essere configurata per funzionare con due diversi formati di frame: il *formato standard*, o base, (descritto in CAN 2.0 A e CAN 2.0 B) e il *formato esteso* (descritto solo da CAN 2.0 B). La distinzione tra il formato base e il formato esteso viene effettuata utilizzando il bit IDE, trasmesso come dominante nel primo caso e come recessivo nel secondo. Tutti i frame iniziano con un bit di *Start-Of-Frame* (SOF) che indica l'inizio della trasmissione del messaggio.

Il protocollo presenta quattro tipi di frame:

Data frame

Sono i frame che eseguono l'effettiva trasmissione dei dati. I messaggi possono avere due formati: *Base frame format* (necessario) o *Extended frame format* (facoltativo).

Il protocollo *base* permette $2^{11} = 2048$ tipi di messaggi diversi, ma da specifiche Bosch se ne possono usare solo 2031. Nella versione *extended* si possono avere fino a $2^{29} = 536.870.912$ tipi di messaggi.

Base frame format Caratterizzati da 11 bit di identificazione, presentano la struttura riportata nella Figura 3.5:

	Field name	Length (bits)	Purpose
	Start-Of-Frame (SOF)	1	Indica l'avvio della sequenza di trasmissione
IDENTIFIER	Identifier	11	Identificatore (univoco) dei dati
	Remote Transmission Request (RTR)	1	Dominante (0) per i Data Frame e recessivo (1) per i Remote Frame
	Identifier Extension bit (IDE)	1	Dominante (0) per i Data frame in formato base
	Reserved bit	1	Riservato
	Data Length Code (DLC)	4	Numero di byte di dati (0-8)
	Data field	0 - 64	Dati da trasmettere (la lunghezza è specificata dal campo DLC)
	Cyclic Redundancy Check (CRC)	15	Controllo di ridondanza ciclico
	CRC delimiter	1	Recessivo (1)
	ACKnowledgement	1	Il trasmettitore invia un bit recessivo e ogni ricevitore può confermare la ricezione con un bit dominante
	ACK delimiter	1	Recessivo (1)
	End-Of-Frame (EOF)	7	Recessivo (1)

Figura 3.5: Struttura Data frame - formato base.

Un vincolo imposto al campo dell'identificatore è che i primi 7 bit non possono essere tutti recessivi.

Extended frame format In Figura 3.6 viene riportata la struttura di tali frame:

	Field name	Length (bits)	Purpose
	Start-Of-Frame (SOF)	1	Indica l'avvio della sequenza di trasmissione
IDENTIFIER	Identifier A	11	Prima parte dell'identificatore (univoco) dei dati, rappresenta anche la priorità del messaggio
	Substitute remote request (SRR)	1	Recessivo (1)
	Identifier Extension bit (IDE)	1	Recessivo (1) per i Data frame in formato esteso
	Identifier B	18	Seconda parte dell'identificatore (univoco) dei dati, rappresenta anche la priorità del messaggio
	Remote transmission request (RTR)	1	Dominante (0) per i Data Frame e recessivo (1) per i Remote Frame
	Reserved bits	2	Riservato
	Data Length Code (DLC)	4	Numero di byte di dati (0-8)
	Data field	0 - 64	Dati da trasmettere (la lunghezza è specificata dal campo DLC)
	Cyclic Redundancy Check (CRC)	15	Controllo di ridondanza ciclico
	CRC delimiter	1	Recessivo (1)
	ACKnowledgement	1	Il trasmettitore invia un recessivo (1) e ogni ricevitore può confermare la ricezione con un dominante (0)
	ACK delimiter	1	Recessivo (1)
	End-Of-Frame (EOF)	7	Recessivo (1)

Figura 3.6: Struttura Data frame - formato esteso.

I due identificatori (A e B) combinati, formano un unico identificatore di 29 bit.

Remote frame

Generalmente la trasmissione dei dati viene eseguita su base autonoma, con il nodo trasmittente che invia un Data frame. È anche possibile, tuttavia, che un nodo di destinazione richieda i dati all'origine, inviando un Remote frame.

Sono identici ai Data frame, se non per:

- Bit RTR posto allo stato recessivo;
- DLC indica il numero di Byte di dati del messaggio richiesto.

In caso di trasmissione contemporanea di un Data frame ed un Remote frame aventi lo stesso identificatore, il primo vince l'arbitrato grazie al bit RTR dominante che segue l'Identifier.

Error frame

Viene inviato nel momento in cui un nodo rileva un errore al fine di renderlo noto all'intera rete, che collaborerà alla rilevazione fornendo la propria visione.

L'Error frame è composto da due campi:

- La combinazione dei flag di errore attivati da uno dei nodi della rete;
- Il cosiddetto *delimitatore di errore* o *Error Delimiter*.

Esistono due tipi diversi di Error Flag:

- *Active Error Flag*: trasmessi da un nodo che ha rilevato un errore di rete, e che si trova nello stato di *Error Active*;
- *Passive Error Flag*: trasmessi da un nodo che ha rilevato la presenza sulla rete di un Active Error Flag, e che si trova nello stato di *Error Passive*.

Overload frame

Utilizzato quando si necessita introdurre un ritardo tra Data Frame e/o Remote Frame. Esistono due condizioni di overload che possono determinare la trasmissione di un flag di questo tipo:

- *Problemi di ricezione*: un ricevitore richiede un ritardo di trasmissione dal successivo Data frame o Remote frame;
- *Contesa del canale*: viene rilevato un bit dominante durante la trasmissione.

È composto da due campi: *Overload Flag* e *Overload Delimiter*. Il primo è costituito da sei bit dominanti (000000). Il secondo è costituito da 8 bit recessivi (11111111) ed ha la stessa forma di un Error Delimiter.

3.3.2 ID allocation

I Message ID all'interno di una rete CAN devono essere obbligatoriamente univoci, in caso contrario, infatti, due nodi continuerebbero a trasmettere oltre la fine del campo di arbitrato (ID) causando un errore.

All'inizio degli anni '90, la scelta degli ID veniva fatta semplicemente sulla base dell'identificazione del tipo di dati e del nodo trasmittente. Tuttavia, poiché l'ID viene utilizzato anche come priorità del messaggio, ciò comportava scarse prestazioni nelle applicazioni real-time. In questi scenari, veniva comunemente richiesto un sottoutilizzo del canale (circa 30%) per garantire le scadenze di tutti i messaggi.

Nelle moderne implementazioni, invece, gli ID vengono determinati in base alla scadenza del messaggio: più corta è la scadenza, minore è l'ID numerico e maggiore è la priorità del messaggio. Così facendo, si riesce ad aumentare l'utilizzo del canale fino al 70-80% senza riscontrare fault dovuti al mancato rispetto delle scadenze.

3.3.3 Bit timing

Tutti i nodi sulla rete CAN devono funzionare alla stessa bit-rate, ma rumore, spostamenti di fase ed altri fattori fanno si che la velocità effettiva possa non essere uguale a quella nominale. Poiché non viene utilizzato un segnale di clock separato, è necessario un mezzo per sincronizzare tutti i nodi. La sincronizzazione è fondamentale durante l'arbitrato, poiché in questo momento i nodi devono essere in grado di leggere contemporaneamente i dati trasmessi da loro stessi e i quelli trasmessi degli altri nodi.

La sincronizzazione inizia sul primo cambio di stato da recessivo a dominante che si presenta dopo un periodo di inattività del canale, e viene rieseguita su ogni transizione di questo tipo che si presenta all'intero del frame. Il Controller CAN si aspetta che il cambio di stato avvenga in un multiplo del *nominal bit time*, ma se la transizione non si verifica in quell'esatto momento, allora il Controller regola di conseguenza tale tempo.

La regolazione si ottiene dividendo ciascun bit in un numero di intervalli di tempo chiamati quanti e assegnando un numero di quanti a ciascuno dei quattro segmenti all'interno del bit: *sincronizzazione*, *propagazione*, *segmento di fase 1* e *segmento di fase 2*. Un esempio di tale frammentazione è riportato nella Figura 3.7:

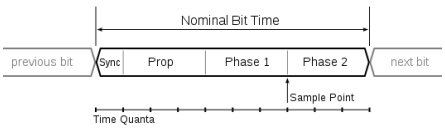

Figura 3.7: Frammentazione di un bit.

Il numero di quanti in cui è diviso il bit può variare in base al controller e il numero di quanti assegnati a ciascun segmento può essere variato in base alla velocità di trasmissione e alle condizioni della rete.

Una transizione che si verifica prima o dopo il previsto porta il Controller a calcolare la differenza di tempo e allungare il segmento di fase 1 o accorciare il segmento di fase 2 di questo tempo. Ciò regola efficacemente i tempi del ricevitore rispetto al trasmettitore per sincronizzarli. Come già anticipato, questo processo di sincronizzazione viene eseguito continuamente ad ogni transizione da recessiva a dominante, per garantire la sincronizzazione tra trasmettitore e ricevitore. In questo modo si riducono gli errori indotti dal rumore a cui il canale è soggetto.

3.3.4 ACK slot

Lo *slot di conferma* (ACKnowledge slot) viene utilizzato per confermare la ricezione di un frame valido. Ogni nodo che riceve il frame senza rilevare un errore, trasmette un livello dominante nello slot ACK e quindi ignora il livello recessivo del trasmettitore. Se un trasmettitore rileva un livello recessivo in questo slot, capisce che nessun ricevitore ha trovato un frame valido. Un nodo ricevente può quindi trasmettere un livello recessivo per indicare che ha ricevuto un messaggio invalido, ma un altro nodo che ha ricevuto un frame valido è libero di sovrascriverlo con un dominante. In questo modo, il nodo trasmittente non può sapere se il messaggio è stato ricevuto da tutti o alcuni nodi della rete.

Spesso, la modalità di funzionamento del dispositivo prevede la ritrasmissione ripetuta dei frame non riconosciuti, schema che può però portare prima allo stato *error passive*, in cui il non distrugge attivamente il traffico del bus quando rileva un errore (non inoltra un livello recessivo), ed in seguito a *bus off*, ovvero che il nodo non partecipa affatto al traffico del bus

3.3.5 Interframe spacing

Data frame e Remote frame sono separati dai messaggi precedenti da un campo chiamato *Interframe Space*, costituito da almeno tre bit recessivi. Quando viene rilevato un bit dominante, questo si considera come *Start-Of-Frame* del messaggio immediatamente successivo.

3.3.6 Bit stuffing

Questa pratica, che consiste nell'inserire un bit di valore opposto dopo cinque bit consecutivi dello stesso valore, è necessaria a causa della codifica utilizzata nel frame, di tipo NRZ (Non Return to Zero), che in caso di valori consecutivi uguali mantiene lo stesso valore di tensione e non genera transizioni utili a risincronizzare i dispositivi

comunicanti. I frame sottoposti a questa operazione vengono poi "decodificati" dal ricevitore, che rimuove i bit precedentemente inseriti. Di conseguenza, quando vengono ricevuti sei bit consecutivi dello stesso valore (111111 oppure 000000), essi vengono considerati un errore.

3.4 CAN Database Files

I file di database CAN [26] sono file di testo, generalmente riservati, che contengono definizioni del segnale e informazioni di ridimensionamento per il contenuto del messaggio CAN. Vengono utilizzati da alcuni fornitori per facilitare la comprensione finale del traffico dati in particolari applicazioni (principalmente automotive). Di seguito i dati che sono normalmente archiviati in tali database:

- Nome del canale;
- Posizione (bit iniziale) e dimensione (numero di bit) del canale all'interno di un messaggio;
- Tipo di dato (signed, unsigned e IEEE float);
- Scaling factor;
- Range;
- Valore di default;
- Commento.

È possibile utilizzare queste informazioni per convertire facilmente i dati di un frame CAN (byte) in un valore comprensibile all'uomo, generalmente relativi ad una grandezza fisica. La Figura 3.8 riporta un semplice esempio di tale conversione (i dati di ridimensionamento necessari sono contenuti nel database):

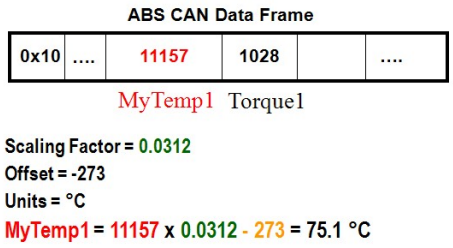

Figura 3.8: Esempio di utilizzo dei dati contenuti nel CAN Database File.

Un database CAN può quindi contenere le definizioni di frame e di segnale per un intero veicolo.

3.5 Low-level ISO/SAE standards - Physical Layer

Il protocollo CAN presenta diversi livelli fisici utilizzabili, i quali classificano alcuni aspetti della rete come livelli elettrici, impedenza del mezzo trasmissivo, velocità di trasmissione massima e altri fattori strettamente dipendenti dal tipo di applicazione da soddisfare. Gli strati fisici più comuni e utilizzati, net tempo convertiti in standard, sono riportati di seguito.

3.5.1 La serie ISO 11898

Specifica il livello fisico e di collegamento (livelli 1 e 2 del modello ISO/OSI [25]) del protocollo CAN, che supporta multiplexing e controllo distribuito real-time per l'uso all'interno di veicoli stradali.

ISO 11898-1:2015 – Data link layer and physical signaling [32]

Descrive l'architettura generale di CAN in termini di strati gerarchici secondo il modello di riferimento ISO/OSI [25] e fornisce le caratteristiche per impostare uno scambio di informazioni digitali tra i moduli che implementano il DLL (Data Link Layer [27]) CAN con specifica dettagliata dei sublayer di controllo logico del collegamento (Logical Link Control, LLC [28]) e di controllo di accesso al canale (Media Access Control, MAC [29]).

ISO 11898-2:2016 – High-Speed CAN [33]

Layer fisico più utilizzato in applicazioni automotive ed industriali. Specifica l'unità di accesso al canale (Medium Access Unit, MAU [30]) ad alta velocità (trasmissione fino a 1 Mbit/s) e alcune features di interfacciamento dipendenti dal mezzo di trasmissione (Medium Dependent Interface, MDI [31]). La specifica richiede l'utilizzo di una coppia di cavi intrecciati (Twisted Pair) come mezzo trasmissivo.

ISO 11898-3:2006 – Low-Speed/Fault-Tolerant CAN [34]

Specifica un'interfaccia a bassa velocità, tollerante ai guasti, dipendente dal mezzo di trasmissione, per l'implementazione di uno scambio dati tra centraline elettroniche di veicoli stradali equipaggiati con CAN a velocità di trasmissione tra 40 kbit/s e 125 kbit/s.

ISO 11898-4:2004 – Time-triggered CAN [35]

Impone le specifiche pe l'implementazione di una comunicazione time-triggered del protocollo CAN (TTCAN). È applicabile alla creazione di uno scambio temporizzato di dati tra le unità di controllo elettronico (ECU) dei veicoli stradali dotati di CAN e specifica l'entità di sincronizzazione del frame che coordina il funzionamento sia del collegamento logico (LLC) che dei controlli di accesso al mezzo (MAC).

ISO 11898-5:2007 – High-Speed/Low-Power CAN [36]

Rappresenta un'estensione della norma ISO 11898-2, che tratta delle nuove funzionalità per i sistemi che richiedono funzionalità a basso consumo energetico qualora non vi sia attività sul canale.

3.5.2 ISO 11992-1:2003 – F/T for road vehicles CAN [37]

Specifica lo scambio di informazioni digitali tra veicoli stradali con massa totale massima autorizzata superiore a 3.500 kg. Comprende le direttive per la comunicazione tra veicoli rimorchiati, in termini di parametri e requisiti dello strato fisico e di collegamento. Include anche test di conformità dello strato fisico.

3.5.3 SAE J2411 - Single-Wire CAN [38]

Definisce i requisiti del livello fisico per qualsiasi collegamento dati *Carrier Sense Multiple Access/Collision Resolution* (CSMA/CR) che utilizza un singolo conduttore come mezzo di comunicazione per lo scambio dati tra le unità di controllo elettronico (ECU) sui veicoli stradali.

3.5.4 Flexible Data-Rate CAN [39]

Utilizzato principalmente nei moderni veicoli ad alte prestazioni, CAN FD è un'estensione del protocollo originale rilasciato nel 2012 da Bosch. Sviluppato per soddisfare la necessità di aumentare la velocità di trasferimento dei dati nelle moderne ECU, a differenza del suo predecessore permette di variare dinamicamente data-rate (fino a 5 volte superiore rispetto allo standard precedente) e dimensione dei messaggi, che possono contenere fino ad un massimo di 64 byte di dati.

3.6 Protocolli di alto livello CAN-based

Poiché lo standard CAN non include le attività dei protocolli del livello applicazione, come il controllo del flusso, l'indirizzamento dei dispositivi e il trasporto di blocchi di dati più grandi di un messaggio, sono state create molte implementazioni di protocolli di livello superiore. Per alcune aree di business sono stati creati diversi standard, ma nel settore automotive ogni costruttore usa utilizzare il proprio protocollo.

3.6.1 CANopen

Lo standard CANopen [22] è costituito da uno schema di indirizzamento, diversi piccoli protocolli di comunicazione e un livello applicazione definito da un profilo del dispositivo specifico. I protocolli di comunicazione supportano la gestione della rete, il monitoraggio dei dispositivi e la comunicazione tra nodi, incluso un semplice livello di trasporto per la segmentazione/desegmentazione dei messaggi. Il protocollo di livello inferiore che implementa il collegamento dati ed i livelli fisici è, in genere, il comune CAN.

3.6.2 Universal Measurement and Calibration Protocol (XCP)

XCP [40] è un protocollo di rete sviluppato come successore del CAN Calibration Protocol (CCP), per il collegamento di sistemi di calibrazione a centraline elettroniche (ECU). Permette l'accesso in lettura e scrittura alle variabili ed al contenuto della memoria di sistemi microcontrollori in fase di esecuzione: interi set di dati possono essere acquisiti o modificati in modo sincrono rispetto agli eventi innescati da timer o condizioni operative. Il protocollo supporta, inoltre, la programmazione della memoria flash.

3.7 Conclusioni

Dall'analisi fatta fino a questo momento risulta evidente come il protocollo CAN sia stato di fondamentale importanza per tutto lo sviluppo tecnologico che ha caratterizzato il settore automotive e l'industria in generale. Il considerevole data-rate raggiungibile, il semplice ed economico mezzo trasmissivo, la programmazione flash, le capacità di rilevamento degli errori e la predisposizione a lavorare anche in ambienti ostili sono solo alcune delle features che caratterizzano lo standard e hanno si che esso diventasse una scelta obbligata per svariate applicazioni.

D'altro canto, però, il protocollo presenta alcune restrizioni che ne limitano l'uso in contesti *smart* generici. Segue una rapida overview delle principali problematiche:

- Limite massimo di 64 nodi della rete, non specificato dallo standard (dovuto al carico elettrico del canale);
- Lunghezza massima (nominale) del canale di 40m;
- Frequenti interazioni inaspettate tra i nodi (ad es. per la gestione degli errori);
- Necessità di spese per sviluppo e manutenzione del software;
- Necessaria impostazione topologica volta a limitare lo stub network (traffico verso l'esterno obbligato ad attraversare il NAT della rete);
- Necessità di corrette terminazioni sui nodi terminali per ridurre i problemi di integrità del segnale;
- Necessario riposizionamento/aggiunta di terminazioni del canale in caso di rimozione di nodi.

Tutto ciò, in aggiunta all'evoluzione generale del networking, favorisce l'integrazione di sistemi/protocolli diversi sulla stessa piattaforma, veicolo o macchina industriale che sia, per permettere l'introduzione di tecnologie sempre più all'avanguardia, che risultano quindi costose non solo in termini finanziari.

Comprensibile, a questo punto, la presa di posizione sempre più ingombrante di IVN eterogenee, nelle quali si prospetta, come si vedrà dai prossimi capitoli, che a dominare saranno le comunicazioni IP-based.

Capitolo 4

IP-based In-Vehicle Networks

Nei moderni veicoli un gran numero di unità di controllo (ECU) è collegato da diversi bus di comunicazione (specifici per il settore automobilistico), con l'obiettivo di facilitare applicazioni distribuite innovative. Allo stesso tempo, computer e dispositivi di intrattenimento utilizzano la tecnologia di comunicazione IP per connettersi ad Internet, consentendo soluzioni avanzate e dando spazio a rapidi cicli di evoluzione.

Nei veicoli di prossima generazione, quindi, molte applicazioni dovranno beneficiare della maggiore larghezza di banda che Ethernet può offrire, rendendo così le reti veicolari basate su IP (IP-based IVN) un componente chiave dei sistemi di trasporto intelligenti.

Nel seguente capitolo viene introdotto il concetto di IP-based IVN, affrontando una veloce panoramica dello sviluppo temporale di Ethernet nel settore automotive e della comune terminologia che si usa quando si parla di queste reti.

Verranno presentate le architetture e gli standard attualmente utilizzati per implementare reti veicolari, ed in seguito si analizzeranno nel dettaglio concetti quali autoconfigurazione dell'IP, routing, gestione della mobilità, servizi DNS e tanto altro.

Il capitolo si concluderà quindi con una sintesi finale dei vantaggi e delle problematiche introdotte con queste tecnologie.

4.1 Overview

Le tradizionali IVN, come visto nel capitolo introduttivo, comprendo diverse tecnologie di rete come CAN e FlexRay, ognuna implementata come soluzione ottimizzata per casi d'uso specifici. Solo da un paio d'anni alcune grandi case automobilistiche hanno iniziato ad includere Ethernet nelle loro auto, per due principali tipi di applicazioni:

- *Diagnosi e aggiornamenti*: considerando la sempre crescente quantità di dati che circolano nel veicolo, diventa chiaro che l'accesso tradizionale basato su CAN è troppo lento per aggiornare, ad esempio, i dati del sistema di navigazione, che possono raggiungere facilmente un paio di Gigabyte di dimensione; l'aggiunta di Ethernet a 100 Mb/s alla presa di diagnosi (OBD-II) è stata quindi un modo semplice ed economico per fornire la larghezza di banda richiesta a soddisfare questo tipo di necessità;

- *Intrattenimento*: utilizzato, ad esempio, per collegare il sistema di intrattenimento dei sedili posteriori (*Rear Seat Entertainment*, RSE) all'unità principale e per fornire un accesso ad alta velocità all'archiviazione di massa in essa contenuta; Ethernet si rivela quindi la soluzione perfetta anche per l'implementazione di un file system di rete, essendo nato proprio per tali applicazioni.

Sono stati questi casi d'uso a rendere possibili le prime implementazioni di Ethernet in un veicolo, ma ad oggi è lecito considerare questa tecnologia anche per altre applicazioni di maggior rilievo, i cui requisiti sono in qualche modo più elevati. Tra queste:

- Collegamento di videocamere digitali avanzate (ad es. LIDAR [41]);

- Sostituzione dei principali protocolli attualmente utilizzati per infotainment e scambio dati;

- Comunicazione tra centraline con maggior capacità computazionale, ad esempio per le future applicazioni di assistenza alla guida (ADAS [5]).

Attualmente sono tre le diverse topologie (Figura 4.1) maggiormente utilizzate per implementare le reti di bordo: *a stella*, *daisy-chain* o una *combinazione di entrambe* (che genera una struttura ad albero). Queste permettono di sviluppare una rete Ethernet senza alcuna modifica agli standard IEEE 802.3 [42] e IEEE 802.1Q [43].

- *A stella*: riduce la complessità del cablaggio, quindi l'installazione e i costi di manutenzione;

- *Daisy-chain*: configurazione più semplice, con switch a 3 porte e apparecchiature per auto *fixed*;

- *Ad albero*: buon compromesso tra prestazioni e costi di manutenzione;

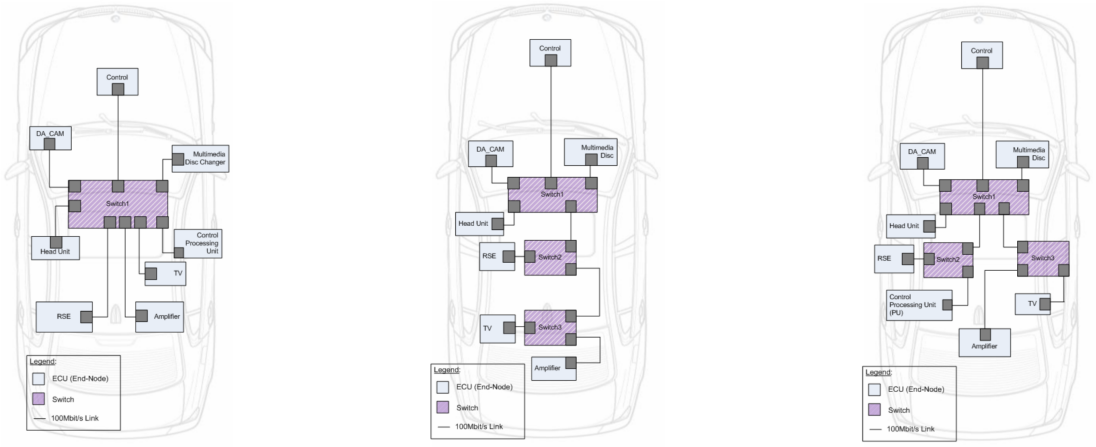

Figura 4.1: Topologie per IVN (da destra: a stella, daisy-chain, ad albero).

4.1.1 Terminologia

Road-Side Unit (RSU)

Nodo che dispone di un dispositivo DSRC (*Dedicated Short Range Communications*) [6] per comunicazioni wireless con veicoli. È anche connesso a Internet come router o switch per l'inoltro di pacchetti.

On-Board Unit (OBU)

Nodo che dispone di un dispositivo DSRC per comunicazioni wireless con altre OBU e RSU. Per rendere più efficiente la navigazione, normalmente è incluso in un veicolo un ricevitore per la radio-navigazione (ad es. *Global Positioning System* (GPS)) comprensivo di OBU;

Vehicle Detection Loop (o Loop Detector)

Dispositivo induttivo utilizzato per rilevare veicoli che passano o arrivano ad una certa posizione (ad es. avvicinamento ad un semaforo). Solo masse metalliche superiori ad una determinata dimensione sono in grado di innescare il rilevamento.

Traffic Control Center (TTC)

Nodo che archivia le informazioni sull'infrastruttura stradale, le statistiche del traffico (ad es. velocità media delle vetture) e le informazioni sui veicoli (ad es. identificatore, posizione, direzione, velocità e percorso). Può comunicare con i nodi

dell'infrastruttura stradale per condividere i dati di misurazione e le informazioni di gestione tramite un protocollo a livello di applicazione.

Wireless Access in Vehicular Environments (WAVE)

Standard IEEE per servizi di sicurezza alla guida basati sul DSRC, attraverso i quali i protocolli IPv6 e Mobile IP possono essere estesi alle reti veicolari.

4.2 Casi d'uso

4.2.1 Vehicle to Infrastructure (V2I)

Comprendono applicazioni quali:

- *Servizi di navigazione*: tramite la rete V2I, tali servizi riescono ad interagire con il TCC per l'ottimizzazione del traffico stradale globale, e possono guidare i singoli veicoli in tempo reale su percorsi di navigazione appropriati (fornendo, ad esempio, i percorsi più rapidi a i veicoli di emergenza);
- *Notifica di incidenti*: implementabile tramite reti RSU o 4G-LTE, permette la notifica immediata di un incidente alle autorità competenti, che vengono prontamente informati su luogo, dinamica e dimensione;
- *Servizi di protezione dei pedoni*: l'uso della rete V2I può agevolare la riduzione di collisioni tra veicoli e pedoni, che tramite il proprio smartphone vengono informati dalla rete stradale sul traffico circostante.

4.2.2 Vehicle to Vehicle (V2V)

Comprendo applicazioni essenziali per la guida autonoma, tra cui:

- *Context-Aware Safety Driving (CASD) navigator*: può aiutare i conducenti a guidare in sicurezza, facilitando il riconoscimento di oggetti invisibili e situazioni pericolose in tempo reale, attraverso la rete V2V;
- *Cooperative Adaptive Cruise Control*: aiuta i veicoli ad adattare autonomamente la propria velocità, attraverso la comunicazione V2V, in base alla mobilità dei loro predecessori/successori in una strada urbana o extraurbana;
- *Platooning*: consente ad una serie di veicoli (ad es. un gruppo di camion) di muoversi insieme ad una distanza molto ravvicinata; in questo modo, i camion possono utilizzare la comunicazione V2V, oltre ai sensori onboard,

per mantenere uno spazio costante tra di loro (da 3 metri a 10 metri); ciò può massimizzare il rendimento del traffico veicolare in un'autostrada e ridurre il consumo di carburante (e quindi l'inquinamento), poiché il veicolo principale può aiutare i seguenti ad affrontare una minore resistenza dell'avanzamento.

4.3 Architetture esistenti

4.3.1 VIP-WAVE: IP nelle reti veicolari 802.11p

Cespedes [63] ha proposto un IP veicolare per WAVE, chiamato VIP-WAVE, per reti I2V e V2I. Lo standard WAVE non supporta il *Duplicate Address Detection* (DAD) dell'*IPv6 Stateless Address Autoconfiguration* (SLAAC), comunicazioni senza interruzioni per i servizi Internet e le comunicazioni multi-hop tra un veicolo ed un nodo dell'infrastruttura (ad es. RSU). Per superare queste limitazioni VIP-WAVE implementa tre strategie:

- DAD e meccanismo efficiente per l'assegnazione dell'indirizzo IPv6;
- Mobilità IP on-demand basata su *Proxy Mobile IPv6* (PMIPv6);
- Comunicazioni *one-hop* e *two-hop* per reti I2V e V2I.

In WAVE, il protocollo IPv6 *Neighbor Discovery* (ND) non è raccomandato a causa del sovraccarico di ND rispetto alle comunicazioni tempestive delle reti veicolari. Tramite il *WAVE Service Advertisement management frame* (WAS), una RSU può fornire ai veicoli le informazioni sulla configurazione IP (ad es. prefisso IPv6, lunghezza prefisso, gateway, tempo di vita del router e server DNS) senza utilizzare ND. Tuttavia, i dispositivi WAVE possono supportare il readdressing per fornire pseudonimia, quindi l'indirizzo MAC di un veicolo può essere modificato o generato in modo casuale. Questo aggiornamento dell'indirizzo MAC può portare alla collisione di un indirizzo IPv6 basato su di esso, quindi VIP-WAVE include un ND leggero e on-demand per eseguire DAD.

VIP-WAVE è quindi un buon candidato per networking I2V e V2I, che supporta comunicazioni ottimizzate ND, handover e comunicazioni *two-hop* (tramite un relè).

4.3.2 IPv6 per WAVE

Sebbene l'obiettivo principale di WAVE sia stata la consegna tempestiva di informazioni relative alla sicurezza, oggi viene presa in considerazione come possibile area applicativa anche la distribuzione di applicazioni di intrattenimento basate su IP. Pertanto, al fine di permettere tale traffico, WAVE supporta IPv6 e protocolli di trasporto come TCP e UDP.

Baccelli [65], dopo aver pubblicato una propria analisi dettagliata del funzionamento di IPv6, ha concluso che l'uso dello stack standard previsto dal protocollo, come stabilito dalla famiglia di specifiche IEEE 1609 [44], non è sufficiente. Al contrario, sostengono che l'assegnazione dell'indirizzamento dovrebbe seguire le considerazioni fatte per i modelli di collegamento ad hoc, simili quindi alle caratteristiche del modello WAVE. Ulteriori sfide come il supporto della pseudonimità attraverso la modifica dell'indirizzo MAC e l'idoneità delle applicazioni TCP tradizionali sono attualmente sotto analisi.

4.3.3 Framework multicast per reti veicolari

Questo framework si occupa di due fasi:

- *Inizializzazione* o *Bootstrap*: include un processo di autoconfigurazione geografica multicast e un metodo di definizione del gruppo di appartenenza;
- *Multicast traffic dissemination phase*: include un meccanismo di selezione della rete su cui trasmettere e uno di consegna multicast che dipende dal nodo destinatario.

Gli autori definiscono un meccanismo distribuito che consente ai veicoli di configurare un indirizzo multicast comune: *Geographic Multicast Address Autoconfiguration* (GMAA). Ogni veicolo è in grado di cambiare l'indirizzo multicast a cui è *abbonato* quando cambia la sua posizione.

Per affrontare le sfide dell'autoconfigurazione dell'indirizzo, gli autori propongono un meccanismo di autoindirizzamento multicast geografico distribuito per gruppi di veicoli, nonché un semplice schema di consegna dei dati nelle reti ibride da un server al gruppo di veicoli in movimento.

4.3.4 Reti IP congiunte ad Architetture Radio

L'approccio propone di considerare una topologia IP in modo simile a quelle di collegamento radio (una sottorete corrisponde all'intervallo di comunicazione veicolare a 1 hop) e definisce tre tipi di veicoli: *Leaf Vehicle* (LV), *Range Extending Vehicle* (REV) e *Internet Vehicle* (IV). La prima classe corrisponde alla più grande serie di veicoli comunicanti (o nodi di rete all'interno di un veicolo), mentre la seconda si pone come obiettivo quello di costruire un *relè IP* tra due IP-subnet e due reti sub-IP. Infine, l'ultima classe corrisponde ai veicoli connessi a Internet. Sulla base di queste tre classi si definiscono sei tipi di topologie corrispondenti alla comunicazione V2V tra due LV a portata diretta, due LV su un veicolo che estende la portata, la comunicazione V2I diretta tramite IV, tramite un IV secondario, o tramite un REV che si collega a sua volta ad un IV.

Un altro aspetto analizzato è quello di unire il livello IP con i canali di collegamento radio. Si propone quindi di separare diverse sottoreti in diversi canali WiFi/ITS-G5, che potrebbero essere *pubblicizzati* nel sistema dal REV. Di conseguenza, l'interferenza globale potrebbe essere controllata all'interno di ciascuna sottorete. Questo approccio è simile alle proposte di gestione della topologia multicanale nelle reti di sensori multi-hop, ma adattato a una topologia IP.

4.3.5 Accesso Mobile ad Internet in FleetNet

Bechler [66] ha descritto l'approccio del progetto FleetNet per integrare l'accesso a Internet nelle future reti veicolari. Il documento FleetNet è probabilmente uno dei primi documenti che affronta questo aspetto e, in molti modi, introduce concetti che verranno successivamente utilizzati in MIPv6 o in altri schemi di gestione della mobilità IP successivi. L'articolo descrive un'architettura V2I composta da veicoli, *gateway Internet* (IGW), *proxy* e *Corresponding Nodes* (CN). Considerando che le reti veicolari richiedono l'utilizzo di indirizzi IPv6, così come la tecnologia di accesso wireless ITS-G5 (nuova in quel momento), una delle sfide principali è quella di interfacciare le due diverse reti (ovvero VANET e Internet IPv4/IPv6). Si introduce quindi un *Fleetnet Gateway* (FGW), che consente ai veicoli in IPv6 di accedere a Internet IPv4 e di collegare due tipi di tecnologie di accesso (network e radio). Un'altra importante sfida è mantenere attivo il routing e i flussi di comunicazione anche mentre i veicoli si spostano tra le FGW. A tal proposito, l'articolo introduce un nodo proxy (un *MIP Home Agent* ibrido), che può reindirizzare i flussi verso il nuovo FGW oltre a fungere da NAT IPv4-IPv6 locale.

FleetNet è stato un documento pionieristico, che ha contribuito a cambiare il MIP e ha portato alla nuova architettura IPv6, attualmente nota come *Proxy-MIP* [46], e il successivo DMM-PMIP.

4.3.6 Architettura a strati per DTN veicolari

Soares [67] ha affrontato il caso della rete veicolare tollerante al ritardo, per le quali, anziché costruire un complesso percorso multi-hop VANET-IP, i veicoli possono essere utilizzati per trasportare i pacchetti più vicino possibile alla destinazione o direttamente alla destinazione stessa. Gli autori hanno costruito l'architettura e il protocollo *DTN Bundle* [67], proponendolo come un'estensione alle VANET (*Vehicular Ad-hoc Networks*) [45] e introducendo tre tipi di nodi: nodi terminali (che richiedono dati), nodi mobili (che trasportano dati lungo i loro percorsi) e nodi di inoltro (memorizzazione dei dati dei mobile nodes agli incroci stradali).

La principale innovazione in questo documento è quella di proporre un'architettura VANET DTN che separa il piano di controllo e il piano dati. Gli autori hanno affermato che è stato progettato per consentire la piena libertà di selezionare la

tecnologia più appropriata, nonché per consentire la comunicazione fuori banda per piccoli pacchetti del piano di controllo e l'uso di DTN in banda per il piano dati.

Sebbene le architetture DTN si siano evolute da quando la relativa specifica è stato scritta, il documento Vehicular-DTN adotta un approccio diverso alla gestione della mobilità IP. Uno dei principali aspetti trattati è quello della separazione del piano di controllo dal piano dati, per consentire una grande flessibilità nella coordinazione di un piano dati eterogeneo con tecnologia di accesso radio (RAT) [47].

4.3.7 Problematiche

V2I-based internetworking

La rete mobile del veicolo e la rete fissa della RSU sono autonome, hanno più sottoreti e dispongono di un router perimetrale per la comunicazione con un altro veicolo o RSU. Il collegamento tra due reti interne tramite comunicazione V2I o V2V richiede uno scambio di dati per ottenere il prefisso di rete ed altri parametri (livello di collegamento, livello MAC e informazioni sul livello IP) per una comunicazione IP tra veicolo ed RSU, o tra due veicoli vicini.

Una volta eseguite le operazioni di individuazione dei parametri di rete e scambio dei prefissi, i pacchetti possono essere trasmessi tra la rete mobile del veicolo e la rete fissa della RSU. Il DNS deve essere in grado di supportare la risoluzione dei nomi per host o server residenti nella rete mobile del veicolo o nella rete fissa della RSU.

V2V-based internetworking

L'assegnazione del prefisso per ciascuna sottorete all'interno della rete mobile di ciascun veicolo viene effettuata tramite un protocollo di delega del prefisso.

4.4 Standard IEEE/ISO in uso

4.4.1 Linee guida IEEE per WAVE – Architettura

IEEE 1609 [44] è una suite di standard per l'accesso wireless in ambienti veicolari (WAVE) sviluppata dalla *IEEE Vehicular Technology Society* (VTS) [48]. Definiscono un'architettura ed un insieme standardizzato di servizi e interfacce che consentono comunicazioni wireless *veicolo-veicolo* (V2V) e *veicolo-infrastruttura* (V2I).

IEEE 1609.0 fornisce una descrizione dell'architettura e delle operazioni del sistema WAVE, chiamato *modello di riferimento WAVE*, che include due stack di protocollo del piano dati (che condividono uno stack comune per i livelli data-link

e fisico): il protocollo Internet standard IPv6 ed il *WAVE Short Message Protocol* (WSMP), progettato per il funzionamento ottimizzato in un ambiente veicolare wireless. I messaggi *brevi*, *WAVE Short Messages* (WSM) possono essere inviati su qualsiasi canale, mentre il traffico IP è consentito solo sui canali di servizio, in modo da scaricare il canale di controllo.

4.4.2 Standard IEEE per WAVE – Servizi di rete

IEEE 1609.3 definisce i servizi che operano a livello di rete e di trasporto, a supporto della connettività wireless tra i dispositivi onboard e quelli fissi a bordo strada che utilizzano *5,9 GHz Dedicated Short-Range Communications* o *Wireless Access in Vehicular Enviroments* (DSRC/WAVE) [49].

I servizi di rete WAVE rappresentano il livello 3 (rete) e il livello 4 (trasporto) dello stack di comunicazione OSI. Lo scopo è quindi fornire servizi di routing all'interno di un sistema WAVE, abilitando più stack di livelli superiori ed inferiori (sopra e sotto WAVE Networking Services). Il supporto dello strato superiore include applicazioni di bordo che offrono sicurezza e praticità agli utenti.

Gli standard WAVE supportano IPv6, che è stato selezionato poiché dovrebbe essere un protocollo utilizzabile nel prossimo futuro.

4.4.3 ETSI ITS: GeoNetwork-IPv6

ETSI ha pubblicato uno standard che specifica la trasmissione di pacchetti IPv6 tramite il protocollo *ETSI GeoNetworking* (GN) [50], definita nella norma ETSI EN 302 636-6-1 usando un sottostrato di adattamento del protocollo chiamato *GeoNetworking to IPv6 Adaptation Sub-Layer* (GN6ASL). Questo consente ad una stazione ITS che esegue il protocollo GN, e un livello di protocollo conforme a IPv6, di:

- Scambiare pacchetti IPv6 con altri ITS-S;
- Acquisire indirizzi unicast IPv6 instradabili a livello globale e comunicare con qualsiasi host IPv6 situato in Internet, tramite la connettività diretta a Internet o tramite altre stazioni ITS di inoltro;
- Eseguire operazioni nelle vesti di *Mobile Router* per la mobilità di rete.

Il documento introduce tre tipi di collegamento virtuale: il primo fornisce la raggiungibilità simmetrica mediante confini stabili geograficamente, mentre gli altri due possono essere utilizzati quando è richiesta la definizione dinamica del dominio di trasmissione. La combinazione di questi tre tipi di collegamento virtuale nella stessa stazione consente di eseguire il protocollo ND IPv6, incluso SLAAC, nonché

di distribuire altro traffico multicast locale e, allo stesso tempo, di raggiungere nodi al di fuori dei confini geografici specifici.

Il documento descrive anche come supportare il *bridging* su GN6ASL e come i pacchetti IPv6 sono incapsulati in pacchetti GN. Specifica, inoltre, il supporto del traffico multicast e anycast IPv6 e la scoperta dei vicini. Per motivi di latenza, lo standard raccomanda vivamente di utilizzare SLAAC per la configurazione dell'indirizzo IPv6.

4.4.4 ISO Intelligent Transport System: IPv6 su CALM

ISO ha pubblicato uno standard che specifica i protocolli e i servizi di rete IPv6 per l'accesso alle comunicazioni per i dispositivi terrestri (*Communication Access for Landing Mobiles*, CALM [51]). Questi servizi sono necessari per supportare la raggiungibilità globale e la connettività Internet continua per ITS-S, nonché la funzionalità di trasferimento richiesta per mantenere tale connettività. Questa funzionalità consente inoltre ai dispositivi legacy di utilizzare efficacemente un ITS-S come router di accesso per connettersi a Internet.

I requisiti si applicano a tutti i tipi di nodi che implementano IPv6: personali, veicolo, lato strada o nodo centrale. Lo standard definisce i moduli funzionali IPv6 necessari in un ITS-S IPv6, che copre l'inoltro e l'interfaccia tra IPv6 e livelli inferiori (ad es. interfaccia LAN), la gestione della mobilità e la sicurezza. Definisce inoltre i meccanismi da utilizzare per configurare l'indirizzo IPv6 per i nodi statici e per i nodi mobili, mantenendo al contempo la raggiungibilità da Internet.

4.5 Configurazione automatica dell'indirizzo IP

4.5.1 Autoconfigurazione dell'indirizzo IP nelle VANETs

Fazio [68] ha proposto una *Vehicular Address Configuration* (VAC) [53] per la configurazione automatica dell'indirizzo IP nelle VANET, che utilizza un protocollo di configurazione host dinamico (DHCP) [52] distribuito. Questo schema utilizza un leader, che svolge il ruolo di server DHCP, all'interno di un cluster con veicoli connessi all'interno di una VANET nella quale il VAC elegge dinamicamente il veicolo leader per fornire rapidamente gli altri di indirizzi IP univoci. Il leader mantiene aggiornate le informazioni sugli indirizzi configurati nella VANET a cui esso è collegato, per ridurre la frequenza di riconfigurazione dovuta alla mobilità.

VAC definisce *scope* come un'area geografica delimitata all'interno della quale gli indirizzi IP sono garantiti come unici. Se il veicolo esce da tale scope, deve richiedere un altro indirizzo IP ad al nuovo leader, in modo che sia di nuovo univoco

nella nuova area. Questo approccio può consentire una modifica meno frequente di un indirizzo IP rispetto all'assegnazione da un gateway Internet fisso.

VAC può supportare quindi l'autoconfigurazione di indirizzi per scenari V2V, ma il sovraccarico per garantire l'unicità degli indirizzi IP non è ignorabile in mobilità ad alta velocità.

4.5.2 Utilizzo delle informazioni di Corsia/Posizione

Kato [69] ha proposto uno schema di assegnazione dell'indirizzo IPv6 utilizzando le informazioni sulla corsia e sulla posizione. In questo schema di indirizzamento, ogni corsia di un segmento di strada ha un prefisso IPv6 univoco. Quando un veicolo si sposta in una nuova corsia, questo configura automaticamente il suo indirizzo IP con il prefisso assegnato a tale corsia.

Tuttavia, questo schema di autoconfigurazione dell'indirizzo potrebbe avere un sovraccarico eccessivo quando i veicoli cambiano spesso corsia segmento di strada.

4.5.3 GeoSAC: autoconfigurazione scalabile dell'indirizzo IP

Baldessari [70] ha proposto uno schema di autoconfigurazione dell'indirizzo IPv6 scalabile, chiamato *GeoSAC*, che utilizza il concetto di rete geografica in modo tale da combinare la funzionalità *IPv6 Neighbor Discovery* (ND) e la funzionalità di routing geografico standard. Nell'IPv6 standard, tutti i nodi all'interno dello stesso collegamento devono comunicare tra loro, ma a causa delle caratteristiche dei collegamenti wireless, questo concetto di non è ben definito nelle reti veicolari. GeoSAC definisce un collegamento come un'area geografica con una partizione di rete, che può avere una VANET ad essa collegata.

Il documento GeoSAC identifica otto requisiti chiave per l'autoconfigurazione dell'indirizzo IPv6 nelle le reti veicolari:

- Configurazione di indirizzi globalmente validi;
- Cassa complessità per l'autoconfigurazione dell'indirizzo;
- Sovraccarico minimo di segnalazione dell'autoconfigurazione;
- Supporto alla mobilità della rete attraverso il rilevamento dei movimenti;
- Efficace selezione del gateway da più RSU;
- Autoconfigurazione completamente distribuita per la sicurezza della rete;
- Autenticazione e integrità dei messaggi di segnalazione;

- Protezione della privacy degli utenti dei veicoli.

Per supportare il concetto di collegamento proposto, GeoSAC esegue il routing ad-hoc per reti geografiche in un livello sub-IP chiamato *Car-to-Car (C2C) NET*. I veicoli all'interno dello stesso collegamento possono ricevere un messaggio di annuncio del router IPv6 (RA), in modo che questi possano autoconfigurare il loro indirizzo in base al prefisso IPv6 contenuto nell'AR ed eseguire il *Duplicate Address Detection* (DAD) per verificare l'univocità dell'IP configurato con l'aiuto del routing geografico all'interno del collegamento.

GeoSAC può così supportare il concetto di collegamento IPv6 attraverso l'instradamento all'interno di un'area geografica specifica.

4.5.4 Problematiche

Esistono due approcci per l'indirizzamento IPv6 nelle reti veicolari: il primo prevede l'utilizzo di indirizzi IPv6 locali unicast univoci, mentre l'altro richiede di utilizzare indirizzi IPv6 globali per l'interoperabilità con Internet. Il primo approccio, talvolta utilizzato dalle *Mobile Ad Hoc Networks* (MANET) [54], può supportare il servizio di notifica di emergenza e il servizio di navigazione nelle reti stradali. Tuttavia, per i servizi Internet generali (ad es. accesso alla posta elettronica, navigazione web e servizi di intrattenimento), è richiesto il secondo approccio.

Per gli indirizzi IP globali, vi sono due opzioni: un approccio di sottorete multi-link per più RSU ed un approccio ad unica sottorete. Nel primo caso le RSU svolgono un ruolo di switch di livello 2, rendendo necessaria l'introduzione di un router che memorizza la posizione di ciascun veicolo appartenente ad ogni RSU. Nel secondo caso, invece, ogni RSU svolge il ruolo di un router nella propria sottorete.

Neighbor discovery

Neighbor Discovery (ND) è una parte fondamentale della suite di protocolli IPv6. I veicoli si muovono rapidamente all'interno della copertura di comunicazione di una RSU, perciò i parametri di ND relativi al tempo (come il tempo di vita del router e l'intervallo *Neighbor Advertisement* (NA)) devono essere regolati per i veicoli ad alta velocità e la loro densità. Man mano che i veicoli si muovono più velocemente, l'intervallo NA dovrebbe diminuire affinché i messaggi raggiungano prontamente i veicoli vicini. Al crescere della densità, invece, l'intervallo NA dovrebbe aumentare affinché i messaggi abbiano una probabilità minore di collisione.

IP Address autoconfiguration

Nelle reti V2I la configurazione automatica dell'indirizzo IPv6 potrebbe non funzionare bene, poiché i veicoli possono percorrere l'intera area di copertura della

RSU prima dell'avvenuto completamente dell configurazione degli indirizzi (con le procedure *Router Advertisement* e *Duplicate Address Detection* (DAD)).

Per mitigare l'impatto della velocità del veicolo sulla configurazione dell'indirizzo, la RSU può eseguire l'autoconfigurazione in modo proattivo come proxy ND per conto dei veicoli. Se i veicoli comunicano periodicamente le informazioni di movimento (ad es. posizione, traiettoria, velocità e direzione) a TCC, questo può coordinare le RSU sotto il suo controllo per la configurazione dell'indirizzo IP proattivo dei veicoli con le informazioni relative alla loro mobilità. DHCPv6 [55] può essere utilizzato per la configurazione automatica dell'indirizzo.

4.6 Routing

4.6.1 Valutazione sperimentale di IPv6 su GeoNet

Tsukada [71] ha presentato un lavoro volto a combinare la rete IPv6 ad un protocollo di routing Car-to-Car Network (chiamato *C2CNet*), proposto dal *Car2Car Communication Consortium* (C2C-CC) [56], che utilizza un protocollo di routing geografico. Nell'architettura C2C-CC, il livello C2CNet si trova tra IPv6 e i livelli di collegamento, pertanto un pacchetto IPv6 viene fornito di un'intestazione C2CNet esterna, che introduce la sfida di come supportare i tipi di comunicazione definiti in C2CNet nel livello IPv6.

4.6.2 Location-Aided Gateway Advertisement and Discovery

Abrougui [72] ha presentato uno schema di rilevamento gateway per VANET, chiamato *Location-Aided Gateway Advertisement and Discovery* (LAGAD), che consente ai veicoli di instradare rapidamente i pacchetti verso il gateway più vicino. Il problema principale che LAGAD affronta è determinare il raggio della zona *pubblicitaria* di un gateway, che dipende dalla posizione e dalla velocità di un veicolo.

Un gateway invia perodicamente messaggi pubblicitari (*Gateway Advertisements*, GAdv) ai veicoli vicini. Quando riceve un messaggio di richiesta da un veicolo, il gateway risponde al questo mediante un messaggio di risposta (*Gateway Reply*, GRep), che contiene le informazioni sulla posizione del gateway e il prefisso della sottorete tramite il quale il veicolo di origine può inviare dati.

Il veicolo di origine avvia il processo di individuazione del gateway inviando pacchetti di richieste di routing incapsulati in pacchetti *Gateway Reactive Discovery* (GRD) o in messaggi GReq da inoltrare ai veicoli circostanti. Il GRD contiene sia informazioni di ricerca e routing, che la posizione e la velocità del veicolo sorgente.

4.6.3 Problematiche

L'autoconfigurazione dell'indirizzo IP deve essere modificata per supportare il networking efficiente, mentre il concetto di collegamento IPv6 può essere supportato dal routing geografico per connettere veicoli con lo stesso prefisso IPv6.

IPv6 ND dovrebbe inoltre considerare la frammentazione temporanea della rete, che causa *buchi* nella comunicazione tra i veicoli.

4.7 Gestione della mobilità

4.7.1 VANETs con Network Fragmentation

Quando le velocità dei veicoli sono diverse, le comunicazioni tra di essi potrebbero non funzionare correttamente a causa della frammentazione della rete. Chen ha affrontato questo problema, producendo un documento che propone un protocollo in grado di posticipare il tempo di rilascio degli indirizzi IP sul server DHCP e selezionare un modo più veloce per ottenere il nuovo indirizzo del veicolo.

Il documento afferma che, sebbene le soluzioni già esistenti per la mobilità degli indirizzi possano ridurre i ritardi di consegna, queste non possono funzionare correttamente su VANET, specialmente in caso di frammentazione della rete, circostanza in cui imessaggi non riescono ad essere trasmessi ai veicoli destinatari. Quando si presenta questo fenomeno potrebbe verificarsi anche un aumento della latenza di handoff ed un tasso di perdita di pacchetti più elevato. L'obiettivo principale di questo studio è, quindi, quello di migliorare i lavori esistenti proponendo un nuovo protocollo di mobilità per VANET, con gestione della frammentazione della rete.

Il documento ipotizza che quando un veicolo si sposta in una nuova sottorete, questo riceve un pacchetto dalla stazione base di destinazione (BS) ed esegue quindi la procedura di trasferimento. Tale procedura comprende due parti: il passaggio di livello 2 (nuovo canale di frequenza) e il passaggio di livello 3 (un nuovo indirizzo IP). La procedura di handoff, invece, gestisce la rilevazione del movimento, la procedura DAD e la registrazione. Nel caso di IPv6, la procedura DAD richiede molto tempo e può causare la disconnessione del collegamento.

Nel documento si propone anche un secondo meccanismo di handoff, che prevede le seguenti fasi:

- *Raccolta delle informazioni* (ciascun nodo mobile (veicolo) trasmette periodicamente posizione, velocità e direzioni dei veicoli ad esso vicini);
- *Acquisizione veloce dell'IP*;
- *Cooperazione intraveicolare*;

- *Route redirection.*

4.7.2 Hybrid Centralized/Distributed Mobility Management

Nguyen [73] ha proposto una gestione ibrida della mobilità centralizzata/distribuita, chiamata H-DMM, per supportare veicoli *altamente mobili*. I sistemi *legacy* di gestione della mobilità non sono adatti per scenari ad alta velocità, poiché viene imposto un ritardo di registrazione proporzionale alla distanza tra un veicolo e la sua rete di ancoraggio. H-DMM è progettato per soddisfare requisiti quali tempi di interruzione del servizio, ritardo *end-to-end*, costi di consegna dei pacchetti e costi di tunneling.

H-DMM propone un nodo centrale, chiamato *Central Mobility Anchor* (CMA), che svolge il ruolo di *Local Mobility Anchor* (LMA). Quando un veicolo entra nell'area di un router di accesso mobile (*Mobile Access Router*, MAR) ottiene un prefisso secondo il protocollo DMM legacy. Inoltre, ottiene anche un secondo prefisso dal CMA per un dominio PMIPv6. Ogni volta che attua uno scambio dati tra le sottoreti di due MAR adiacenti, il veicolo mantiene il prefisso del LMA mentre aggiorna quello del MAR. Per un nuovo scambio di dati con un nuovo CN, il veicolo può selezionare il prefisso MAR o quello LMA, ma se il numero di prefissi attivi è maggiore di certa una soglia, allora deve obbligatoriamente usare quello LMA come prefisso per l'indirizzo di origine. Inoltre, esso può continuare a ricevere pacchetti di dati destinati agli indirizzi IPv6 precedenti, tramite il protocollo DMM legacy.

H-DMM può quindi supportare un tunneling efficiente per un veicolo in movimento ad alta velocità tra le sottoreti di due MAR adiacenti. Tuttavia, quando H-DMM richiede di eseguire DAD per il test di unicità dell'indirizzo configurato nella sottorete del successivo MAR, l'attivazione di tale indirizzo verrà ritardata.

4.7.3 Architettura ibrida per il Network Mobility Management

Nguyen [74] ha proposto H-NEMO, un sistema ibrido di gestione centralizzata della mobilità distribuita per gestire la mobilità IP dei veicoli in movimento. Il supporto di base del *Network Mobility* (NEMO) [60] presenta uno schema centralizzato per la mobilità di rete che eredita gli svantaggi di Mobile IPv6, come il routing non ottimale e il sovraccarico di segnalazioni in scenari *nidificati*, nonché problemi di affidabilità e scalabilità. Al contrario, schemi distribuiti come il DMM (*Distributed Mobility Management*) consentono il supporto della mobilità solo ai flussi di traffico che lo richiedono esplicitamente. Tuttavia, nei veicoli in movimento ad alta velocità, il DMM può soffrire di latenza e costi di segnalazione elevati.

L'architettura H-NEMO proposta non è progettata per una specifica tecnologia wireless. Al contrario, definisce un'architettura ed un protocollo di segnalazione del tutto generale, consentendo inoltre l'uso di DMM o Proxy Mobile IPv6 (PMIPv6) a seconda delle caratteristiche di flusso dati. Per l'allocazione dell'indirizzo IP, il router mobile (MR), o il nodo mobile (MN) collegato a un MR con NEMO, ottiene due set di prefissi: uno dal Central Mobility Anchor e uno dal router di accesso mobile (MAR). In questo modo, MR/MN può scegliere un prefisso più stabile per i flussi di lunga durata ed un prefisso MAR per quelli di breve durata. Lo scenario multi-hop è considerato sotto il concetto di un NEMO nidificato.

Nguyen non ha fornito simulazioni del sistema, ma solo una valutazione analitica che ha considerato i costi di segnalazione e consegna dei pacchetti e che ha dimostrato che H-NEMO supera le precedenti proposte. Per alcune misure, come il costo di segnalazione, H-NEMO può essere più costoso degli schemi centralizzati quando la velocità del nodo aumenta, ma si comporta meglio in termini di costi di consegna dei pacchetti e ritardo di consegna.

4.7.4 NEMO-Enabled Localized Mobility Support

Gli autori del *NEMO Localized Mobility Support* (NEMO-LMS) hanno proposto un'architettura per abilitare la mobilità IP utilizzando uno schema basato su rete PMIPv6, dove solo i terminali mobili sono dotati di mobilità IP. A differenza della mobilità basata su host, PMIPv6 sposta la segnalazione sul lato della rete, in modo tale che il gateway di accesso mobile (*Mobile Access Gateway*, MAG) sia incaricato di rilevare la connessione/disconnessione del nodo.

Soto ha proposto supporto NEMO in PMIPv6 (N-PMIP), schema in cui la funzionalità del MAG è estesa al router mobile (MR), chiamato anche MAG mobile (mMAG). La funzionalità del terminale mobile rimane pressoché invariata, se non che ora esso può ricevere anche un prefisso IPv6 appartenente al dominio PMIPv6. Pertanto, in N-PMIP, il terminale mobile si collega all'MR come se si stesse collegando a un MAG fisso e l'MR si collega al MAG fisso utilizzando la segnalazione PMIPv6. Quando il terminale mobile esegue il roaming su un nuovo MAG o un nuovo MR, la rete inoltra i pacchetti attraverso l'LMA. N-PMIP definisce quindi una funzionalità estesa nell'LMA, che consente una ricerca ricorsiva: innanzitutto si individua la voce di associazione corrispondente a mMAG, dopodiché si individua quella corrispondente al MAG fisso, ed infine si consente all'LMA di inviare i pacchetti al mMAG a cui è attualmente collegato il terminale mobile.

Le prestazioni di N-PMIP sono state valutate mediante simulazioni e confrontate poi con uno schema NEMO + MIPv6 + PMIPv6: i risultati vedono come *vincitore* N-PMIP.

4.7.5 Mobilità per le VANETs

Chen [76] ha proposto un protocollo di mobilità di rete per ridurre il ritardo di trasferimento e garantire la connettività Internet ai veicoli in movimento in autostrada. In questo lavoro, i veicoli possono acquisire indirizzi IP da altri veicoli attraverso le comunicazioni V2V. Nel momento in cui il veicolo esce dalla copertura della stazione base, un altro può aiutarlo in roaming ad acquisire un nuovo indirizzo IP. Le auto sulla stessa corsia sono inoltre autorizzate ad assistere il veicolo nell'esecuzione di un *pre-handoff*.

Gli autori hanno ipotizzato che la connettività wireless sia fornita dalle reti di accesso WiFi e WiMAX [57] ed hanno preso in considerazione scenari in cui un singolo veicolo, ad esempio un autobus, possa aver bisogno di due router mobili per avere un'efficace procedura di pre-handoff. Non è però stata menzionata l'applicabilità del sistema in altri scenari.

4.7.6 Integrazione di VANETs e Fixed IP Networks

Peng [77] ha proposto un nuovo schema di gestione della mobilità per l'integrazione di VANET e reti ad IP fissi. Lo schema proposto riguarda la mobilità dei veicoli sulla base di una struttura stradale anziché di una rete ad-hoc bidimensionale. Utilizza le informazioni fornite dalle reti veicolari per ridurre le spese generali di gestione della mobilità e consente a più stazioni base vicine ad un veicolo di rilevare simultaneamente la connessione ad esso. Ciò porta ad un miglioramento della connettività e della consegna dei dati, senza l'introduzione di messaggi ridondanti. Le prestazioni sono state valutate utilizzando un simulatore del traffico stradale chiamato SUMO (*Simulation of Urban Mobility*) [58].

4.7.7 Gestione della mobilità su base SDN nelle reti 5G

Nguyen [75] ha esteso i suoi precedenti lavori su un DMM adattato in cui considera un'architettura *Software-Defined Networking* (SDN), osservando i vantaggi reciproci che il DMM IPv6 potrebbe ottenere da un'architettura SDN. Nell'architettura proposta viene utilizzato un PMIP-DMM, dove MF è l'OFS per il piano dati, mentre uno o più controller SDN gestiscono il piano di controllo. L'analisi ed il prototipo proposti nel documento dimostrano che tale architettura può fornire una scalabilità superiore rispetto al DMM standard.

Il documento SDN-DMM formula diverse osservazioni che accreditano la gestione della mobilità IP basata su architettura SDN, che sarà sicuramente integrata nelle reti 5G nel prossimo futuro. Dopo aver separato gli indirizzi *Identity* e *Routing*, la gestione della mobilità IP richiede ulteriormente di separare il controllo dal piano dati se questo deve rimanere scalabile per le VANET. Inoltre, il routing basato sul

flusso (in particolare lo standard *OpenFlow* [59]) sarà richiesto nelle future reti veicolari eterogenee e l'SDN, accoppiato con DMM, offre un vantaggio in termini di rilevamento/riconfigurazione del flusso dinamico e di ottimizzazione del percorso più breve.

4.7.8 Mobilità IP per le VANETs: sfide e soluzioni

Cespedes [64] ha fornito un'analisi sulle sfide del supporto di base NEMO per VANET, il quale consente la gestione di un gruppo di nodi (una rete mobile) anziché di un singolo nodo. Tuttavia, sebbene un veicolo o un insieme di veicoli possano essere visti come un gruppo di nodi, NEMO non è stato progettato tenendo conto delle particolarità delle VANET.

Cespedes riassume quindi per prima cosa i requisiti della gestione della mobilità IP, come potenza del dispositivo finale, handover, complessità o consumo di banda ridotti. VANET inoltre aggiunge altri requisiti, come la segnalazione minima per l'ottimizzazione del percorso (*Road Optimization*, RO), la separabilità del flusso, la sicurezza e la protezione della privacy. Come già osservato, tutti questi fattori introducono nuove sfide per la mobilità IP e NEMO BS per VANET.

Si descrivono quindi vari schemi di ottimizzazione disponibili per NEMO BS. Considerando una connessione a hop singolo per CN, una delle principali strategie di ottimizzazione è quella di evitare la deviazione HA e raggiungere direttamente la CN. In questa direzione, vengono proposte alcune ottimizzazioni, come la creazione di un tunnel IP diretto tra MR e CR, l'implementazione di un meccanismo di delega che consente ai nodi in visita di utilizzare direttamente MIPv6 anziché NEMO o, infine, l'ottimizzazione intra-NEMO di un percorso diretto bypassando gli HA.

Generalmente, quando si apre una rotta multi-hop tra una VANET ed un CN, le maggiori sfide si hanno sul fronte della sicurezza e della privacy. Il multi-hop eterogeneo in una VANET (ad es. basandosi su varie tecnologie di accesso) corrisponde infine ad un'altra impegnativa sfida anche per il sistema NEMO BS.

4.7.9 Problematiche

L'analisi affrontata illustra un supporto per la mobilità IP nella rete V2I. In una sottorete di RSU, i veicoli attraversano continuamente le rispettive coperture di comunicazione adiacenti, e durante questo *incrocio* le sessioni TCP/UDP possono essere mantenute attraverso le tecniche di supporto alla mobilità IP, come MIPv6, Proxy MIPv6 e *Distributed Mobility Management* (DMM). Con i rapporti periodici contenenti le informazioni sui movimenti dei veicoli, TCC può coordinare RSU e altri componenti della rete sotto il suo controllo per la gestione proattiva della mobilità dei veicoli lungo il movimento dei veicoli. Tali informazioni, però, devono tenere in considerazione la velocità di movimento dei veicoli, che spostandosi

rapidamente tra i campi di azione delle RSU possono causare un sovraccarico della rete.

Per supportare la mobilità IP è possibile utilizzare anche il protocollo NEMO (*Network Mobility Basic Support Protocol*) [60], ma, come nel caso di MIPv6, la velocità dei veicoli deve essere considerata per una configurazione efficace dei parametri.

Il design della soluzione ottimale per il *Mobility Management* (MM) varia a seconda degli scenari: autostrada o strada urbana. Gli schemi *ibridi* (NEMO + PMIP, PMIP + DMM, ecc.) di solito mostrano prestazioni migliori rispetto agli schemi *puri*, anche se la maggior parte di tali schemi è stata testata solo a livello di simulazione o analitico.

4.8 Servizi DNS

4.8.1 DNS Multicast

Il DNS multicast (mDNS) consente ai dispositivi nell'intervallo di comunicazione a 1 hop di risolvere reciprocamente il nome DNS nel corrispondente indirizzo IP. Ogni dispositivo ha un resolver DNS e un server DNS. Il primo genera una query DNS per l'applicazione del dispositivo, mentre il secondo risponde a tale query.

4.8.2 DNS Name Autoconfiguration per dispositivi IoT

DNS Name Autoconfiguration (DNSNA) propone un servizio di denominazione DNS per dispositivi *Internet-of-Things* (IoT) in una rete su larga scala.

Il servizio prevede quattro passaggi: la generazione del nome DNS, il rilevamento della duplicazione di tale nome, la registrazione del nome e il recupero dell'elenco dei nomi DNS.

Innanzitutto, DNSNA consente a ciascun dispositivo IoT di generare il proprio nome DNS con un suffisso (acquisito da ND o DHCP) e le informazioni sul dispositivo (ad es. fornitore, modello e numero di serie).

In secondo luogo, ciascun dispositivo controlla se il suo nome DNS è in conflitto con quello di qualche altro dispositivo nella stessa sottorete.

Dopodiché, ogni dispositivo registra il proprio nome DNS e il corrispondente indirizzo IPv6 in un server DNS designato tramite un router, il quale raccoglie periodicamente le informazioni dei dispositivi presenti nelle sue sottoreti.

Infine, un utente può recuperare l'elenco dei nomi DNS dei dispositivi disponibili all'utente tramite il server DNS designato. Una volta che l'utente recupera tale elenco può monitorare e controllare in remoto un dispositivo.

4.8.3 Problematiche

La risoluzione dei nomi DNS prevede la traduzione dei nomi DNS nel corrispondente indirizzo IPv6, possibile attraverso un server DNS ricorsivo (RDNSS) interno alla rete del veicolo ed un server DNS in Internet, che si trovano all'esterno del VANET. Gli RDNSS possono essere pubblicizzati tramite DNS RA o DNS DHCP nelle sottoreti della vettura.

mDNS è progettato per piccole reti ad-hoc, con raggio di comunicazione wireless/cablato ad 1 hop. Se viene utilizzato in reti veicolari con più sottoreti non può è in gradi di lavorare in modo efficace, poiché il messaggio di query di ciascun resolver DNS deve essere multicast in tutta la rete mobile, portando quindi ad un grande volume di traffico DNS. DNSNA è invece progettato per una rete su larga scala con più sottoreti, perciò se viene utilizzato nella rete di un veicolo con più sottoreti riesce a funzionare comunque efficacemente.

4.9 Service Discovery

4.9.1 mDNS-based Service Discovery

Noto protocollo basato su DNS (*DNS-based Service Discovery*, DNS-SD) che utilizza un *DNS Service Resource Record* per supportare il rilevamento dei servizi forniti da un dispositivo o da un server. Un RR SRV contiene: nome di istanza del servizio, nome del servizio, protocollo a livello di trasporto, nome di dominio, numero di porta e il nome DNS del dispositivo idoneo per il servizio richiesto. Con questo DNS-SD, un host può cercare un'istanza di servizio attraverso SRV RR per ottenere un elenco di dispositivi corrispondenti al tipo di servizio cercato.

4.9.2 ND-based Service Discovery

L'ND veicolare si propone come estensione del ND IPv6 per il rilevamento del prefisso e del servizio. I veicoli e le RSU possono annunciare i propri prefissi e servizi nella loro rete interna tramite messaggi ND, che contenenti tutte le informazioni necessarie. Dal momento che non si richiede alcun protocollo aggiuntivo nel livello applicazione, questo approccio permette a veicoli ed RSU un rilevamento rapido dei prefissi e dei servizi di rete.

4.9.3 Problematiche

I veicoli devono scoprire i servizi (ad es. notifica delle condizioni stradali, servizi di navigazione e di intrattenimento) forniti dai nodi dell'infrastruttura tramite RSU.

Durante il passaggio di un incrocio o di un tratto stradale in cui vi è un RSU, i veicoli devono eseguire rapidamente questo processo.

DNS-SD basato su mDNS può essere utilizzato per tale rilevamento tra veicoli, o tra un veicolo ed una RSU, utilizzando un protocollo multicast. Questo introduce però un ritardo non trascurabile, poiché il messaggio di individuazione del servizio deve attraversare tutta la rete in modalità multicast.

Un approccio possibile è l'utilizzo di un *Piggyback Service Discovery* durante lo scambio dei prefissi tra la rete mobile e la rete fissa. In questo caso, lo scambio di prefissi include informazioni sul servizio, come l'indirizzo IP di ciascun nodo, il protocollo del livello di trasporto utilizzato e il numero di porta. L'ND veicolare è in grado di supportare questo approccio in modo efficiente.

4.10 Sicurezza e Privacy

4.10.1 Protezione delle comunicazioni IPv6

Fernandez [78] ha proposto uno schema di comunicazione IPv6 veicolare sicuro utilizzando *Internet Key Exchange versione 2* (IKEv2) e *Internet Protocol Security* (IPsec). Questo schema mira al supporto sicuro di *IPv6 Network Mobility* (NEMO) per i dispositivi di bordo tramite un router mobile (*Mobile Router*, MR), che normalmente dispone di più interfacce wireless (GSM, IEEE 802.11p, WiFi e WiMAX [57]). L'architettura proposta è costituita da: una *Vehicle ITS Station* (nodo con una rete mobile e un MR), una *Roadside ITS Station* (RSU utilizzata come gateway per connettere reti veicolari ad Internet) ed una *Central ITS Station* (TCC, dotato di MR, utilizzato come agente domestico (*Home Agent*, HA) per la gestione della posizione dei veicoli).

Lo schema di comunicazione proposto imposta sessioni sicure IPsec per il controllo ed il traffico dei dati tra il MR della Vehicle ITS Station e l'HA della stazione centrale. La stazione a bordo strada svolge invece un ruolo di Access Router (AR) per fornire la connettività Internet alle stazioni veicolari tramite interfacce wireless, come IEEE 802.11p, WiFi e WiMAX. Nel caso in cui la Roadside ITS Station non sia disponibile, il veicolo comunica direttamente con la stazione centrale tramite rete cellulare (ad es. 4G).

Lo schema di comunicazione sicura migliora quindi il protocollo NEMO che interagisce con IKEv2 e IPsec per la mobilità di rete nelle IVN.

4.10.2 Autenticazione e Controllo degli accessi

Moustafa [79] ha proposto un sistema di sicurezza che fornisce servizi di autenticazione e autorizzazione nelle reti veicolari e che mira al supporto di servizi

sicuri ed affidabili per il traffico dati. Lo schema garantisce un trasferimento di dati confidenziale tra le parti comunicanti (ad es. nodo veicolo e infrastruttura) utilizzando IEEE 802.11i (ovvero WPA2 [61]).

Gli autori hanno proposto un'architettura di rete veicolare composta da tre entità: la rete di accesso, le *Wireless Mobile Ad-hoc Networks* (MANETs) e gli *Access Point* (AP). La rete di accesso è l'infrastruttura di rete fissa, mentre le MANET sono costituite dai veicoli. Gli AP, infine, formano l'infrastruttura WLAN IEEE 802.11 che costituisce l'interfaccia tra la rete di accesso e le MANET.

Per i servizi AAA (*Authentication, Authorization, and Accounting*), l'architettura proposta utilizza un modello di autenticazione *Kerberos* [62], che autentica i veicoli interfacciati all'AP e li autorizza quindi ad accedere a vari servizi. Poiché i questi vengono autenticati una sola volta, lo schema di sicurezza proposto può ridurre al minimo il carico sull' *Authentication Server* (AS) e ridurre il ritardo imposto dall'utilizzo di IEEE 802.11i.

4.10.3 Problematiche

Sicurezza e privacy sono fondamentali nelle reti V2I e V2V, che solo i veicoli autorizzati dovrebbero poter utilizzare per la comunicazione con server esterni e tra dispositivi a bordo di vetture diverse.

Un numero di identificazione del veicolo (*Vehicle Identification Number*, VIN) e un certificato utente possono essere utilizzati per autenticare entrambe le parti attraverso un nodo di infrastruttura stradale (una RSU connessa ad un server di autenticazione in TCC). Anche i certificati TLS (*Transport Layer Security*) possono essere utilizzati per le comunicazioni sicure tra veicoli.

Per una comunicazione V2I *sicura*, è necessario stabilire un canale *sicuro* tra il router mobile del veicolo ed il router fisso della RSU, mentre nel caso di comunicazioni V2V è ovviamente necessario farlo tra i router dei due veicoli.

La sicurezza delle reti veicolari dovrebbe fornire, in modo efficiente, servizi di autenticazione ed autorizzazione ai veicoli, considerando non solo il passaggio *orizzontale*, ma anche quello *verticale* (poiché i veicoli hanno più interfacce wireless).

Per impedire ai malintenzionati di rintracciare un veicolo tramite il suo indirizzo MAC o l'IP, è necessario che questi aggiornino periodicamente tali indirizzi, operazione che non deve però interrompere le comunicazioni tra veicolo ed RSU.

4.11 Analisi generale

Alcune case automobilistiche hanno iniziato ormai da tempo ad utilizzare Ethernet per l'implementazione di IP-based IVN che permettono, a differenza della tradizionale rete CAN, l'interconnessione ad alta velocità tra le unità di controllo

elettroniche. Le tecnologie a guida autonoma, inoltre, sono ad oggi sviluppate da molti produttori e società IT e, poiché richiedono l'interazione ad alta velocità tra veicoli, nodi dell'infrastruttura e cloud, le relative reti saranno obbligatoriamente basate su IP.

Ciò richiede, quindi, che le tecnologie dei componenti chiave per le reti veicolari debbano essere importanti ambiti di ricerca, su cui le aziende devono investire per soddisfare le esigenze future e permettere l'implementazione di un'architettura di rete veicolare efficiente.

Attraverso l'analisi fatta si è decretato che la rete veicolare basata su IPv6 può essere ben allineata con gli standard IEEE WAVE per varie applicazioni automotive, come sicurezza di guida, guida efficiente e intrattenimento. Tuttavia, poiché tali standard non raccomandano l'utilizzo di un determinato protocollo di ND, per comunicazioni efficienti in mobilità ad alta velocità è necessario adattare i sistemi di ND.

Per le reti basate su IP, la configurazione automatica dell'indirizzo IP è una funzione preliminare. Poiché i veicoli possono comunicare in modo intermittente con TCC tramite RSU e comunicazioni V2I, TCC può svolgere un ruolo di server DHCP per l'allocazione di indirizzi IPv6 univoci. Questa attività centralizzata permette quindi di rimuovere il ritardo della procedura DAD, necessaria per la verifica dell'unicità degli indirizzi IP.

Per il routing e la gestione della mobilità, la maggior parte dei veicoli è ormai dotata di navigatore GPS, fruibile come sistema dedicato o tramite app per smartphone. I veicoli possono quindi condividere la loro posizione attuale e la propria traiettoria (cioè il percorso di navigazione) con TCC tramite l'uso di questo navigatore, che può così prevedere in anticipo le future posizioni di ciascuna vettura. Con tale previsione, TCC supporta le RSU per eseguire proattivamente l'instradamento e la consegna dei pacchetti di dati.

Capitolo 5

Gateway per IVN

Come anticipato negli scorsi capitoli, il numero sempre crescente di bus di comunicazione differenti presenti all'interno di uno stesso veicolo e la consistente mole di dati scambiati all'interno di tali reti hanno reso essenziale la separazione della funzione di gateway, fino a poco tempo fa integrata nelle comuni ECU, ad un'unità di controllo dedicata ed autonoma.

In questo capitolo viene quindi introdotto il concetto di *Automotive Gateway*, specificandone le principali differenze dai dispositivi generici, le criticità imposte dall'applicazione automotive ed il modello base di elaborazione.

Nella seconda parte del capitolo si affronta invece un'analisi dettagliata di un framework open-source per l'implementazione software di un gateway per IVN [3] basate su CAN, FlexRay ed Ethernet.

5.1 Automotive Gateway

Nati per soddisfare la necessità di collaborazione tra reti veicolari diverse, rappresentano una versione ottimizzata per il mondo automotive dei più generici gateway, e fungono quindi da hub centrale che interconnette e trasferisce in modo sicuro e affidabile i dati attraverso le diverse reti presenti nei veicoli moderni. Forniscono isolamento fisico e traduzione del protocollo per instradare i segnali tra i domini funzionali (trasmissione, telaio, sicurezza, infotainment, telematica, ADAS, ecc.) che necessitano di condivisione di dati per poter raggiungere in modo corretto e completo i propri obiettivi.

In Figura 5.1 viene riportato un comune esempio di utilizzo di un *Automotive Gateway* [80] all'interno di una moderna IVN:

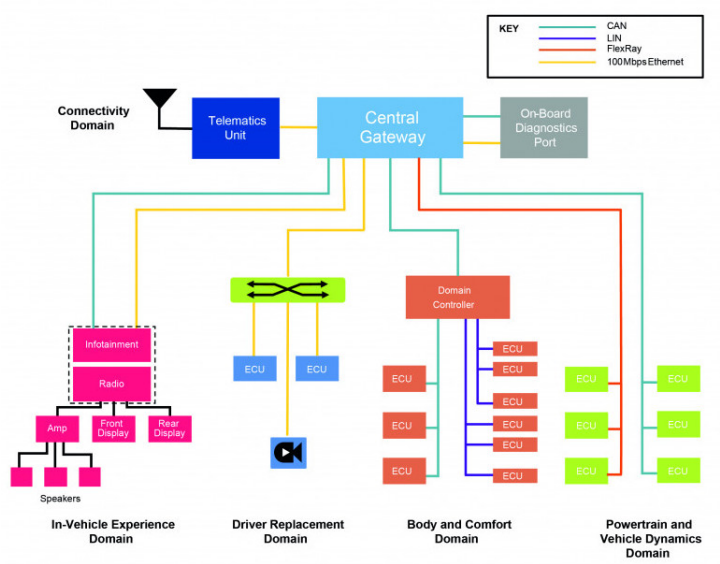

Figura 5.1: Schema di IVN con gateway.

5.1.1 Funzionalità e obiettivi

La funzione principale di un gateway è fornire comunicazioni sicure e senza interruzioni all'interno di una rete, compreso il collegamento tra le svariate reti interne ad un veicolo e quelle all'esterno di esso. Il trasferimento dati senza errori è quindi essenziale per garantire che le ECU dispongano, al momento opportuno, delle informazioni necessarie al corretto funzionamento della vettura. Il gateway deve perciò essere in grado di fornire sempre la comunicazione tra qualsiasi sottorete del sistema, con bassa latenza e jitter [81].

Esistono molte funzionalità del gateway necessarie per permettere comunicazioni senza interruzioni, ma le principali sono:

- *Protocol translation*: tradurre dati e controllare le informazioni da/verso reti di per sé incompatibili per consentire la comunicazione tra di loro;

- *Data routing*: instradamento dei dati per raggiungere la destinazione prevista, che potrebbe trovarsi su una rete diversa da quella di partenza, richiedono così una traduzione di protocollo;

- *Diagnostic Routing*: instradamento di messaggi diagnostici tra dispositivi diagnostici esterni ed ECU, che possono comportare la traduzione tra protocolli diagnostici come DoIP (*Diagnostic-over-IP* [82]) e UDS (*Unified Diagnostic Services* [83]);

- *Firewall*: filtraggio del traffico di rete in entrata e in uscita in base a regole che possono impedire il trasferimento di dati da fonti non autorizzate; i firewall più avanzati possono includere filtri sensibili al contesto;
- *Message mirroring*: acquisizione di dati dalle interfacce di input per la trasmissione su un'altra interfaccia, per diagnostica o archiviazione dei dati;
- *Intrusion detection*: monitoraggio del traffico di rete per individuare anomalie che potrebbero indicare intrusioni;
- *Network management*: gestione degli stati, configurazione della rete e delle ECU ad essa connesse e supporto per la diagnostica;
- *Key management*: elaborazione e archiviazione delle chiavi di rete e dei certificati;
- *OTA management*: gestione degli aggiornamenti firmware OTA (*Over-The-Air*) [84] remoti delle centraline all'interno del veicolo accessibili dal gateway.

5.1.2 Principali criticità

Affrontare i requisiti in rapida crescita del mercato automobilistico per la sicurezza è una sfida sempre più complessa. Le reti automobilistiche possono essere bersaglio di attacchi informatici (in particolare le reti come CAN, che non sono state progettate pensando alla sicurezza) rendendole vulnerabili ai messaggi fantocci e agli attacchi di disturbo del canale. Le interfacce wireless esterne delle auto connesse presentano un altro punto di attacco che aumenta ulteriormente i rischi per la sicurezza. Gli hacker potrebbero estrarre risorse (informazioni private o chiavi crittografiche) o influire sul funzionamento del veicolo sfruttando le vulnerabilità dell'implementazione.

Questi rischi per la sicurezza possono in parte essere contenuti con un *Secure Gateway*, parte di un'architettura di sicurezza a più livelli ideata dalla multinazionale NXP [85], pioniera nel mondo degli Automotive Gateway e della loro sicurezza, che offre un approccio globale a più livelli per la sicurezza automobilistica.

Il livello *Secure Gateway* funge da firewall che controlla l'accesso dalle interfacce esterne (come Internet) alla rete interna del veicolo, e controlla quali nodi nella rete del veicolo possono comunicare tra loro. Il livello di elaborazione sicura (*Secure Processing layer*) implementato dai gateway NXP, invece, garantisce un avvio sicuro e degli schemi di controllo dell'integrità in tempo reale per garantire che il codice sia autentico, affidabile e inalterato; fornisce inoltre un modulo embedded di sicurezza hardware (*Hardware Security Module*, HSM [86]) per la crittografia e la gestione sicura delle chiavi.

I meccanismi di sicurezza proteggono anche le interfacce e le comunicazioni attraverso l'autenticazione dei messaggi, per convalidare i mittenti, la crittografia,

per proteggere l'integrità e la privacy dei dati, e il monitoraggio del traffico, per il rilevamento delle intrusioni e per prevenire pericoli indotti dall'esterno che possono influire sulla sicurezza. È fondamentale quindi che ogni gateway disponga di un ambiente di esecuzione affidabile e che sia fisicamente isolato, con memoria sicura e resistente agli attacchi fisici per mantenere l'integrità della sicurezza.

5.1.3 Flusso di elaborazione

Come largamente anticipato, la funzione principale di un gateway è la traduzione tra protocolli diversi presenti all'interno di uno stesso veicolo (LIN, CAN, FlexRay, Ethernet, ecc.).

Prima di tutto, il dispositivo controlla se è arrivato o meno un nuovo messaggio sulle diverse interfacce. Se è presente un frame, allora viene attivato il task della routine di interruzione (*Interrupt Service Routine*, ISR [87]) per la ricezione del messaggio (la routine specifica dipende dal protocollo), che riceve il messaggio e lo copia nel buffer di memoria. A questo punto, ciascun ISR attiva un'attività di elaborazione dei messaggi tra quelle disponibili per ciascun protocollo, che estrae l'ID, la lunghezza dei dati, nonché i dati stessi dal messaggio ricevuto.

L'attività di elaborazione deve essere a conoscenza dell'interfaccia e dell'ID di output per determinare il protocollo di destinazione. A tale scopo si possono utilizzare principalmente due meccanismi diversi:

- *Direct forwarding*: l'interfaccia e l'ID del messaggio di output sono definiti in modo tale che il gateway trasferisca immediatamente i messaggi; il meccanismo di inoltro diretto richiede però l'elaborazione di un codice per ogni percorso, comportando così un aumento delle dimensioni del codice stesso;

- *Routing table* [88]: utilizza una tabella di routing per determinare sia l'interfaccia che il protocollo del nodo di origine e del nodo di destinazione; tale meccanismo utilizza meno memoria rispetto al *direct forwarding* e permette la modifica a runtime delle tabelle, fattore chiave per la flessibilità di tutto il sistema.

Il gateway procede quindi convertendo il messaggio dal formato di origine a quello di destinazione e memorizzando il frame convertito nello stack del nodo di arrivo.

Infine, prima di procedere con l'effettiva trasmissione dei dati, il dispositivo controlla se il nodo di destinazione è accessibile o meno ed in caso di problemi analizza nuovamente la tabella di routing alla ricerca di un percorso alternativo. Quando non è disponibile alcun percorso alternativo, il gateway sospende la trasmissione del messaggio, che rimane quindi memorizzato nello stack.

La Figura 5.2 riassume il flusso di funzionamento di un comune gateway di questo tipo:

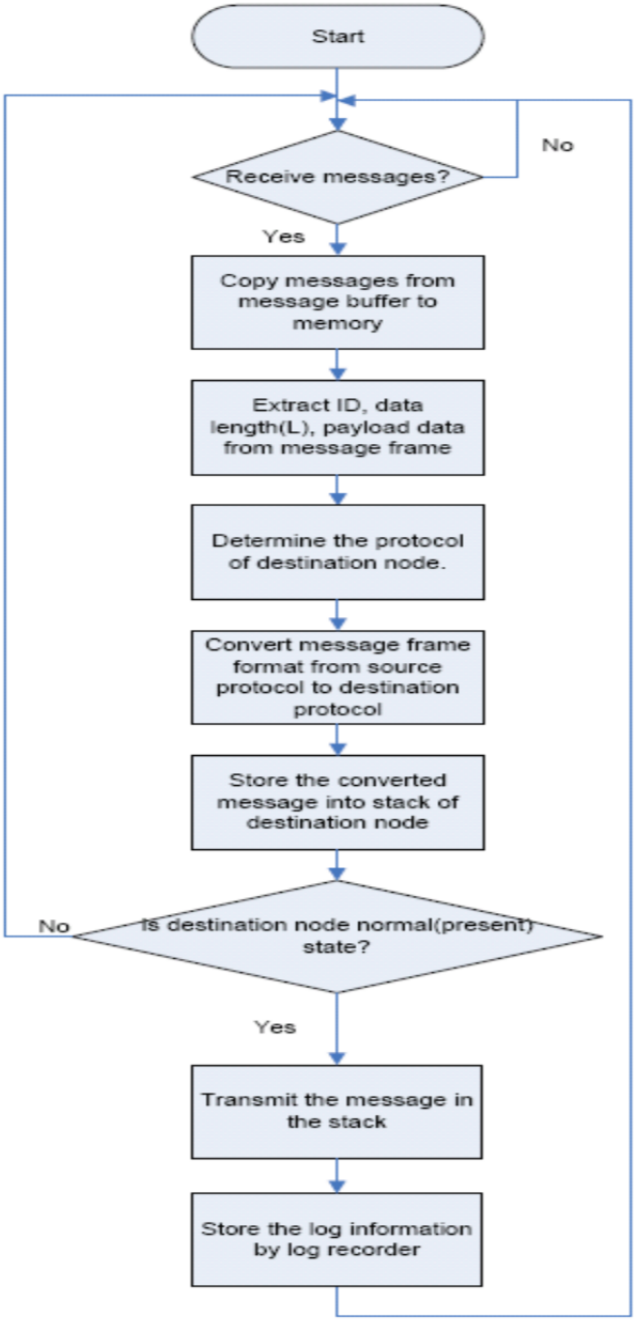

Figura 5.2: Generico flusso di elaborazione di un gateway.

5.1.4 Principali tecnologie in commercio

Seguendo la sempre crescente domanda di moduli robusti ed efficienti in grado di implementare le strategie necessarie per permettere l'integrazione onboard di reti veicolari eterogenee, in gradi di collaborare attraverso lo scambio di dati, alcune grandi multinazionali hanno preso il comando di questo *nuovo* mercato, investendo ormai da tempo importanti risorse dei loro reparti R&D (*Research and Development*).

Bosch – Central Gateway for commercial vehicles (CGW) [89]

CGW, che viene utilizzato come nodo di comunicazione centrale della rete, è la porta per tutti i dati che circolano nel veicolo attraverso vari sistemi di bus diversi (Ethernet, CAN, LIN, ecc.). Funge da router per la comunicazione tra l'interno della vettura e l'esterno, attraverso la *Connectivity Control Unit* (CCU) [90], proteggendo la rete di bordo dall'accesso non autorizzato con strategie di sicurezza come Firewall, *Hardware Security Module* (HSM) [86] o *Intrusion Detection System* (IDS) [91].

FEV – Connected Vehicle Gateway (GATEWAY 5) [93]

Progetto avviato oltre 15 anni fa dalla società tedesca *FEV Motorentechnik GmbH* [92], vede ad oggi la sua applicazione in dozzine di piattaforme di sviluppo e numerosi veicoli commerciali. Il modulo, principalmente progettato per la traduzione di messaggi tra reti veicolari distinte (high-speed CAN, low-speed/fault-tolerant CAN, single-wire CAN, Ethernet, LIN, ecc.), integra numerosi protocolli wireless (BTLE, Wi-Fi, ecc.) ed è in grado di controllare dispositivi di I/O sulla base di messaggi e segnali ricevuti dalle diverse reti a cui esso è collegato.

5.2 Gateway Framework

Rilasciato nell'ottobre del 2015 da un comitato sud-coreano accreditato dall'IEEE [94], si presenta come un ambiente di sviluppo facile da riutilizzare e verificare che è in grado di ridurre costi e tempi di sviluppo per gli ormai essenziali Automotive Gateway [80]. Il framework offre funzionalità all'avanguardia che includono riprogrammazione parallela, routing diagnostico, gestione della rete (*Network Management*, NM), aggiornamento dinamico del routing, configurazione del routing multiplo, sicurezza e tanto altro.

5.2.1 Overview

Come largamente anticipato, il numero di centraline elettroniche (ECU) nei veicoli è aumentato notevolmente negli ultimi anni per consentire una varietà di requisiti e caratteristiche (sicurezza, consumi, infotainment, ecc.). CAN è un protocollo dominante per le IVN, ma non può fornire prestazioni real-time, caratteristica essenziale nelle applicazioni critiche per la sicurezza (sistemi X-by-wire e ADAS). Per risolvere questo problema è stato introdotto FlexRay, utilizzando un meccanismo di accesso multiplo a divisione temporale (TDMA [95]) per sistemi safety-critical. Sebbene questo fornisca una larghezza di banda dieci volte maggiore del CAN, non può però soddisfare le ingombranti richieste di infotainment e multimedia. Ethernet può fornire la larghezza di banda richiesta da questi sistemi e presenta numerosi vantaggi per l'uso nelle IVN, tra cui basso costo, larghezza di banda elevata e teconologia ormai avanzata. Per questo motivo, Ethernet sostituirà probabilmente CAN come protocollo dominante nelle IVN. È così quindi che il gateway diventa un componente di vitale importanza nelle moderne reti eterogenee, senza il quale sarebbe impossibile far frutto della collaborazione dei diversi sottosistemi.

Un importante sviluppo sta caratterizzando anche la comune architettura delle IVN, che passano da *Central Gateway Architecture* a *Backbone-based Architecture*. L'implementazione a *gateway centrale*, mostrata in Figura 5.3, prevede l'utilizzo di un dispositivo centrale che si collega all'intera IVN e fornisce una comunicazione continua tra tutti i protocolli di reti eterogenei. L'architettura basata su *backbone*, mostrata invece in Figura 5.4, prevede l'introduzione di unità di controllo del dominio (*Domain Control Unit*, DCU) che svolgono il ruolo di gateway tra la rete backbone e la propria sottorete.

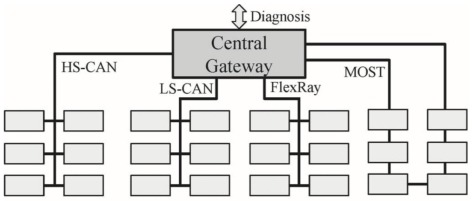

Figura 5.3: Central Gateway Architecture.

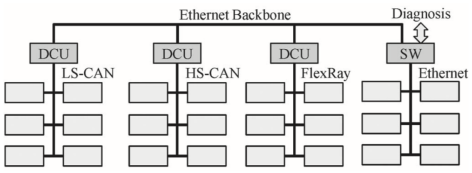

Figura 5.4: Backbone-based Architecture.

Poiché le funzionalità richieste per un gateway di questo tipo sono diverse a seconda del veicolo e le implementazioni finora proposte si sono dimostrate sempre strettamente vincolate da questo fattore, il framework si propone come soluzione in grado di facilitare il riutilizzo e la verifica di questi dispositivi, in maniera completamente indipendente dalle piattaforme hardware e software sottostanti. Presenta alcune principali caratteristiche, tra cui:

- Supporta routing di tipo *frame-based*, *PDU-based* e *signal-based* tra le reti CAN, FlexRay ed Ethernet;
- Supporta il routing diagnostico tra *Diagnostic Internet Protocol* (DoIP) [82] e servizi unificati di diagnostica (*Unified Diagnostic Services*, UDS [83]) e tra UDS che utilizzano protocolli diversi (CAN e FlexRay);
- Permette di riprogrammare parallelamente più ECU, collegate ad interfacce di rete diverse, riducendo l'intero tempo di riprogrammazione;
- È in grado di aggiornare le tabelle di routing e i file di configurazione senza necessità di riprogrammazione dell'intero firmware;
- Permette di memorizzare molteplici tabelle di routing e file di configurazione diversi, supportando così una grande varietà di veicoli;
- Fornisce un sistema di autenticazione per impedire l'accesso non autorizzato e la crittografia/decrittografia per proteggere i dati importanti;
- Può essere configurato e verificato in maniera banale utilizzando un software di configurazione dotato di interfaccia grafica;
- Per essere *sicuro*, deve essere sviluppato utilizzando tecniche di analisi statica per migliorare la qualità del software e deve essere accuratamente verificato. Per un'implementazione completa, dovrebbe inoltre supportare una funzione di callback che può essere programmata dall'utente per la sicurezza in caso di problemi gravi.

5.2.2 Concept

La Figura 5.5 riassume in termini generali il concept dell'ambiente:

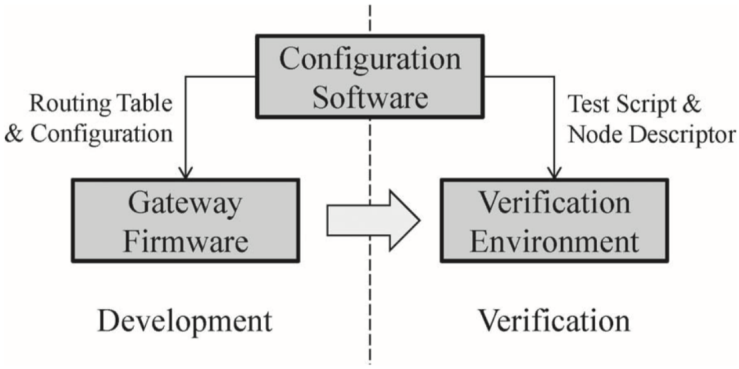

Figura 5.5: Concept del Gateway Framework.

Il framework proposto è costituito quindi da:

- *Configuration Software*: software di configurazione che può generare automaticamente tabelle di routing e configurazioni per il firmware del gateway;
- *Gateway Firmware*: software riutilizzabile che supporta funzionalità gateway all'avanguardia, indipendente dalla piattaforma hardware/software e quindi facilmente trasportabile su differenti centraline. Sebbene il firmware del gateway sia riutilizzabile, le regole di routing e le configurazioni di un gateway differiscono in base al modello del veicolo e ai suoi requisiti, pertanto, le regole e le configurazioni di instradamento devono essere configurate e generate automaticamente di volta in volta, utilizzando il software di configurazione;
- *Verification Environment*: ambiente per la verifica automatizzata del gateway sviluppato, utilizzando script di test e descrittori di nodi virtuali generati dal software di configurazione.

In generale, un gateway può essere sviluppato a partire dal firmware proposto, settato, utilizzando le tabelle di routing e le configurazioni generate dal software di configurazione, ed infine verificato, attraverso l'ambiente di verifica configurato mediante script di test e descrittori di nodi virtuali.

5.2.3 Architettura del framework

Per essere totalmente indipendente dalle piattaforme hardware e software, il framework è stato progettato secondo un'architettura a più livelli che include: *Hardware layer*, *Platform Abstract layer*, *Gateway Framework layer* ed *Application layer*.

La Figura 5.6 mostra l'architettura proposta:

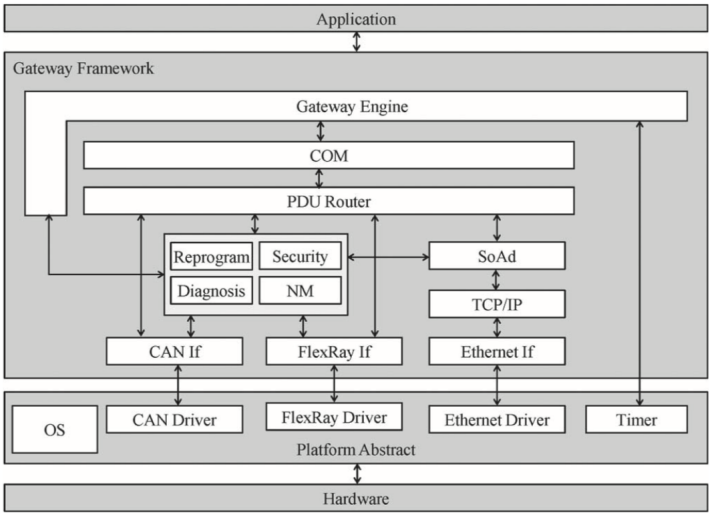

Figura 5.6: Architettura del firmware.

- *Hardware layer*: contiene i componenti hardware come MCU (*Microcontroller Unit*) [96] e controller di comunicazione; si interfaccia solo con il *Platform Abstract layer* della piattaforma e non influisce in nessun modo sugli altri due;

- *Platform Abstract layer*: inserito tra il livello hardware e quello Gateway, è stato introdotto per soddisfare il requisito di indipendenza dalla piattaforma; fornisce driver di dispositivo per controllare il livello hardware e un'interfaccia standard per il livello Gateway; include i driver per CAN, FlexRay, Ethernet e quelli per i timer, nonché i moduli del sistema operativo (OS), ma è possibile escluderli o includerli, a seconda dell'hardware specifico o dei requisiti dell'utente; deve essere sviluppato tenendo in considerazione l'*Hardware layer*, poiché estremamente dipendente dall'hardware specifico;

- *Gateway Framework layer*: è a sua volta compost da:

  - ***Gateway engine***: configura ed esegue i moduli richiesti, secondo le tabelle e le configurazioni di routing, e deve essere periodicamente eseguito dal modulo timer; un'esecuzione anche breve può migliorare le prestazioni del gateway, ma richiede più tempo di elaborazione e deve essere configurato tenendo in considerazione le capacità di elaborazione del processore. Inoltre, controlla periodicamente lo stato degli altri moduli e lancia determinate callback quando si verificano eventi preconfigurati (come timeout della comunicazione o errori hardware);

  - ***Communication stack***: è costituito da un modulo COM, un router PDU, interfacce CAN, FlexRay ed Ethernet, protocollo di controllo della trasmissione TCP/IP e *socket adapter module*;
  - ***Add-on function modules***: includono moduli di diagnosi, riprogrammazione, NM e sicurezza, nonché DoIP, UDS e le relative informazioni di routing. Il modulo di riprogrammazione elabora la richiesta di riprogrammazione contemporanea di diverse ECU per ridurre i tempi di update. Il modulo NM include le strategie di Network Management per OSEK/VDX, AUTOSAR CAN, FlexRay e UDP; il tipo di NM per ciascuna rete è determinato dal software di configurazione. Un modulo di sicurezza fornisce inoltre l'algoritmo di autenticazione e l'algoritmo di encryption/decryption;

- *Application layer*: include un'ampia varietà di *user applications*, sviluppate dagli utenti usando le informazioni e le funzionalità fornite dal *Gateway Framework layer*.

5.2.4 Traduzione e Routing

Il framework proposto presenta quattro tipi diversi di routing: *Frame-based routing*, *PDU-based routing*, *Signal-based routing* e *Diagnostic routing*.

- *Frame-based routing*: il routing basato su frame richiede un processo di traduzione per convertire il formato del frame (compresi intestazione, dati e tutto il resto) in quello del protocollo della rete di destinazione. Il meccanismo genera un nuovo frame per la destinazione, considerando le caratteristiche dei protocolli di rete, come ad esempio la lunghezza massima del frame e il meccanismo di trasmissione (TDMA o *event driven* [95]). Poiché l'obiettivo di questo tipo di routing è instradare quanto più rapidamente possibile i messaggi, esso può essere configurato per lavorare senza passare dalla tabella di routing. Occasionalmente, il routing basato su frame può risultare inefficiente se usato tra protocolli di rete diversi, pertanto viene utilizzato principalmente per l'indirizzamento tra uno stesso protocollo (ad esempio, gateway CAN-to-CAN) e per il routing diagnostico tra dispositivi diagnostici esterni e centraline;

- *PDU-based routing*: le PDU (*Protocol Data Unit*) [97] sono utili per i protocolli di rete che supportano una lunghezza dei frame maggiore rispetto a CAN, come FlexRay ed Ethernet. Il routing basato su PDU può essere diretto e indiretto. Nel primo caso, si ritrasmette una PDU alla rete di destinazione non appena questa viene ricevuta; in questo caso il meccanismo è simile a quello basato su frame, ma può instradare parte del frame (ovvero una

PDU) a una rete di destinazione senza copiare l'intero frame. Il routing indiretto della PDU, invece, copia semplicemente una PDU ricevuta su una PDU di destinazione, la quale potrà essere trasmessa quando soddisferà una determinata condizione di attivazione della trasmissione, in base alla modalità di trasmissione con cui è stata configurata (periodica, mista o diretta). La tabella di routing dei PDU, per ridurre il tempo di ricerca, deve essere ordinata in base al tipo di protocollo e all'ID della rete di origine del messaggio;

- *Signal-based routing*: il routing basato sul segnale instrada i messaggi dalla rete di origine a quella di destinazione secondo una tabella di indirizzamento. Il software di configurazione converte automaticamente il nome del segnale nell'ID del messaggio, che diventa quindi l'identificatore univoco utilizzato dal modulo COM. Il software di configurazione configura, inoltre, le proprietà di comunicazione per il segnale in base al corrispondente database di comunicazione, pertanto la tabella di routing del segnale non include informazioni sulle proprietà di comunicazione. A differenza di altri metodi di routing, questo può modificare tutte le proprietà di comunicazione, inclusi il periodo di trasmissione, la modalità di trasmissione e il valore del segnale, poiché viene elaborato dal task di elaborazione dell'instradamento del segnale, che viene eseguito nello strato superiore del modulo COM e viene periodicamente chiamato dal *Gateway Engine*. Tale task controlla se il segnale sorgente viene man mano aggiornato o meno, leggendo la tabella di routing in sequenza, ed indirizza il segnale sorgente verso la sua destinazione solo se questo è stato aggiornato correttamente. In genere, il task del processo di routing del segnale elabora centinaia di regole di instradamento e richiede tempi di esecuzione lunghi, pertanto l'attività di elaborazione dovrebbe essere fatta in maniera preventiva, per prevenire la perdita di dati ricevuti causata dall'overflow dello stack di input;

- *Diagnostic routing*: Il routing diagnostico inoltra i messaggi tra i dispositivi diagnostici esterni e le ECU, che sono collegati al gateway tramite CAN o FlexRay. DoIP (*Diagnostic-over-IP*) [82] viene utilizzato come protocollo diagnostico per Ethernet, mentre UDS viene utilizzato come protocollo per CAN e FlexRay. Per fornire una comunicazione senza interruzioni tra un dispositivo esterno basato su DoIP e le centraline basate su UDS, il gateway deve ovviamente implementare un processo di traduzione tra questi protocolli. Il gateway controlla il campo *type* nell'intestazione DoIP quando riceve il messaggio: se il valore è quello di un messaggio diagnostico, allora il dispositivo ottiene il valore dell'indirizzo sorgente (*Source Address*, SA) e i campi di indirizzo target (*Target Address*, TA) dal messaggio e cerca le informazioni corrispondenti nella routing table. Se il gateway riesce a risalire

correttamente ai dati necessari all'instradamento, allora memorizza l'ID, che sarà poi utilizzato come identificativo. A questo punto, il gateway crea un messaggio diagnostico per il protocollo della rete di destinazione e lo trasmette: se tale protocollo è CAN, allora il valore ID del messaggio creato è il quello dell'indirizzo target, mentre se il protocollo di destinazione è FlexRay, invece, l'ID del frame deve essere stato determinato alla configurazione.

5.2.5 Riprogrammazione parallela

Un sistema diagnostico che utilizza CAN di comunicazione tra i nodi della rete (le diverse ECU) può aggiornare il software di ogni ECU in sequenza, uno per uno, a causa della larghezza di banda limitata del protocollo. Di conseguenza, è necessario molto tempo per aggiornare tutte le centraline contenute in un sistema automobilistico. Per risolvere questo problema, il DoIP (*Diagnostic-over-IP*) [82] basato su Ethernet viene utilizzato come protocollo diagnostico e, assieme ad un dispositivo di diagnosi esterno collegato ad un gateway, trasmettere le richieste di riprogrammazione a un numero maggiore di ECU della rete. Ciò è possibile poiché Ethernet (minimo 100 Mb/s) fornisce una larghezza di banda molto superiore rispetto a CAN (1 Mb/s) e FlexRay (10 Mb/s).

Poiché l'uso inappropriato degli aggiornamenti del software della centralina costituisce una minaccia per la sicurezza dei sistemi automobilistici, è necessaria quindi l'autenticazione durante la riprogrammazione. Una volta completata la procedura di autenticazione, un dispositivo di diagnosi esterno può trasmettere simultaneamente le richieste di riprogrammazione a tutte le ECU collegate a reti diverse tramite il gateway. Quando il modulo di diagnosi incluso nel gateway riceve una richiesta di update, il framework passa in una modalità di funzionamento in cui esegue il modulo di riprogrammazione. In questa modalità, il gateway interrompe il normale funzionamento e tenta di instradare i messaggi di riprogrammazione il più rapidamente possibile, stabilendo quindi una connessione per ogni richiesta di riprogrammazione, che viene memorizzata in un buffer assegnato a ciascuna connessione.

5.2.6 Network Management

NM è ampiamente utilizzato per controllare gli stati di sospensione e *wake-up* di una ECU e una rete. Il framework fornisce strategie di NM per OSEK/VDX [98], AUTOSAR CAN, AUTOSAR FlexRay e AUTOSAR UDP (per Ethernet) [99]. Il tipo di NM e il relativo eventuale utilizzo per ciascuna rete sono configurati tramite il software di configurazione.

Questi moduli forniscono informazioni sugli stati della rete e sui nodi collegati ad essa, indipendentemente dal fatto che siano in stato di sospensione o di riattivazione.

Ogni regola di routing ha un flag di *wake-up routing* (WU) e può essere configurata dall'utente. Se un flag WU è vero, il gateway trasmette il messaggio di origine alla destinazione dopo la riattivazione di quest'ultima. Nel caso in cui il flag WU sia falso, il gateway ignora il messaggio di origine se la rete di destinazione rimane in uno stato di sospensione.

5.2.7 Configurazione e verifica

Il gateway è ormai diventato parte fondamentale dei sistemi onboard, poiché influisce sulla sicurezza di tutto il sistema automobilistico. Tutti i fattori che possono causare un errore devono quindi essere ridotti al minimo durante il processo di sviluppo e, ad implementazione terminata, il corretto funzionamento del gateway deve essere accuratamente verificato. Il framework proposto fornisce il software di configurazione e l'ambiente di verifica per risolvere questi problemi.

La Figura 5.7 descrive i metodi di configurazione e verifica implementati nel framework:

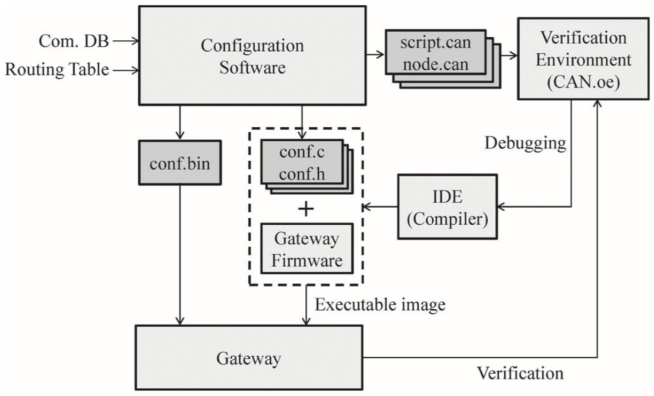

Figura 5.7: Schema di configurazione e verifica del Gateway.

Il software di configurazione è un programma dotato di GUI, pensato per semplificare la configurazione del firmware, che genera i file di configurazione e le tabelle di routing utili per gli script di test e i descrittori di nodi virtuali (necessari per l'ambiente di verifica). Il programma configura automaticamente il gateway leggendo il database di comunicazione, come i CAN DataBase Files, e un file di configurazione del routing, che può essere scritto utilizzando Excel o il formato XML (*Extensible Markup Language*) [100]. Il software genera quindi la configurazione come file binario o come file sorgente, utilizzati successivamente per il debug del sistema e per correggere eventuali errori nati durante il processo di sviluppo del gateway. Un file di configurazione binario può essere scritto in una specifica area

di memoria nel gateway, diversa ovviamente da quella in cui è memorizzato il firmware, che legge quindi tale configurazione prima di inizializzare il dispositivo. Questa funzione, chiamata aggiornamento dinamico del routing, consente all'utente di sviluppare il gateway senza una profonda conoscenza del firmware. In questo modo, il dispositivo può essere riutilizzato per diversi modelli di veicoli, andando a modificare di volta in volta solo le configurazioni, senza dover sviluppare un dispositivo specifico per ciascun modello. Il software può inoltre generare file di configurazione che includono più tabelle di routing diverse, poiché il gateway può utilizzare selettivamente più tabelle. Per ridurre al minimo l'utilizzo della memoria, è necessario comprimere tali tabelle, che anche se vengono compresse non riducono in nessun modo le prestazioni del gateway.

L'ambiente di verifica, come già anticipato, utilizza script di test e descrittori di nodi virtuali per verificare il corretto funzionamento gateway. CAN.oe [102], strumento di monitoraggio e simulazione della rete sviluppato da *Vector* [101], viene utilizzato come ambiente di base per implementare l'ambiente di verifica, che deve replicare lo stesso genere di comunicazione di un veicolo reale, in cui un gateway è collegato ad un elevato numero di ECU. A tale scopo, il software di configurazione genera descrittori di nodi virtuali, ovvero degli script eseguibil che *virtualizzano* una ECU. Sebbene una ECU reale implementi numerosi ed avanzati algoritmi, la versione virtuale si limita solo ad emulare le funzionalità di connettività. È quindi in grado di inviare e ricevere messaggi, proprio come una centralina reale, ma i valori dei messaggi non ovviamente sono *reali*. Per generare i descrittori di nodo vengono utilizzati solo il database di comunicazione, mentre i valori dei dati vengono assegnati in modo casuale a runtime. La Figura 5.8 descrive, in linea generale, il metodo di verifica utilizzato dall'apposito ambiente, che una rete virtuale per ogni singola rete fisica connessa al gateway.

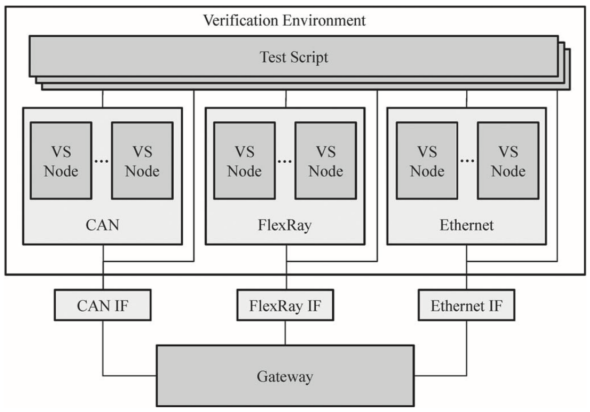

Figura 5.8: Ambiente di verifica del gateway.

I nodi virtuali non includono algoritmi applicativi, che determinano i tempi di trasmissione e il valore dei dati trasmessi. Per risolvere questo problema, viene assegnato un descrittore ad ogni nodo virtuale, che definisce un algoritmo per determinare tempi e valori dei messaggi. Per supportare vari casi di test, tali descrittori supportano varie modalità per ciascun caso di test, che possono essere selezionate prima dell'esecuzione dei test.

5.2.8 Analisi finale

Il framework proposto è utile per un sistema automobilistico attuale che utilizza un gateway centrale, ma sarà ancora più funzionale quando, in futuro, i sistemi automobilistici implementeranno DCU come gateway nelle *Backbone-based Architecture*.

Di seguito si analizzano velocemente i due principali fattori richiesti per un Automotive Gateway [80] che soddisfi le moderne necessità: riusabilità e sicurezza.

Riusabilità

Per migliorarne la riusabilità, il framework è stato progettato utilizzando un'architettura indipendente da hardware e software utilizzati, potendo così essere facilmente trasferito su una larga varietà di piattaforme utilizzando AUTOSAR MCAL [103] come livello astratto della piattaforma. Inoltre, le tabelle e le configurazioni di instradamento, che differiscono per ogni singolo modello di veicolo, sono generate utilizzando un software specifico di configurazione, che elimina quindi la possibilità di errori umani.

Rispetto alle più comuni implementazioni precedenti (standard e non), questo framework offre quindi una riusabilità migliorata fornendo un sistema completo, sicuro ed efficiente.

Sicurezza

Il gateway è un componente software critico, pertanto la garanzia di un elevata qualità del software è importante per la sicurezza dei passeggeri. A tale scopo, sono state utilizzate tecniche di analisi statica per sviluppare il framework, le quali non solo sono in grado di dimostrare che il software soddisfa le specifiche, ma riescono anche a verificare che il questo non presenti violazioni delle procedure di programmazione consigliate, che potrebbero causare comportamenti errati del sistema.

Due differenti tipi di analisi vengono implementati:

- *Motor Industry Software Research Association (MISRA) C coding rules*: MISRA C [104], ampiamente utilizzato nell'industria automobilistica, è un

sottoinsieme di direttive C sicure che impedisce comportamenti pericolosi dei sistemi di controllo integrati;

- *Verifica di potenziali difetti mediante analisi statica*: CodeSonar [105], sviluppato da *Gramma Technology* [106] ed utilizzato in questo caso per implementare questo tipo di analisi, è in grado di rilevare potenziali difetti come concorrenza, puntatori nulli e buffer overflow esaminando il flusso di dati e l'esecuzione simbolica del software.

Poichè il framework proposto fornisce un ambiente di verifica per il gateway sviluppato, risulta quindi necessario anche verificare a sua volta tale ambiente di verifica. A tal proposito, il gateway prototipo viene verificato utilizzando un ambiente di verifica sviluppato da un'organizzazione che produce gateway a livello commerciale.

Il framework non trasmette alcun messaggio a nodi non funzionanti della rete, che non possono quindi ricevere i messaggi. Inoltre, l'ambiente fornisce funzioni di callback per il meccanismo di fail-safety quando vengono rilevati gravi problemi nel sistema.

5.3 Valutazioni complessive

Le auto connesse sono come i dispositivi mobili: sempre connessi e con crescente complessità, prestazioni e requisiti di sicurezza. Le centraline dei futuri veicoli autonomi dovranno collaborare per rilevare, elaborare ed agire secondo i dati rilevati da innumerevoli sensori a bordo del veicolo: tutto ciò, ovviamente, richiede il trasporto e l'elaborazione di un'enorme quantità di dati in modo sicuro tra le unità di controllo.

Come già anticipato, è evidente la tendenza ad adottare la tecnologia multigigabit Ethernet per le IVN, mirando ad utilizzarla come *backbone* per la comunicazione tra domini diversi. Questa transizione vuole distribuire la funzionalità gateway sui controller di dominio (*Domain Controller*), in grado di elaborare e controllare l'instradamento dei dati tra le varie interfacce (le ECU dei domini specifici), mentre un gateway centrale instrada i pacchetti Ethernet tra i domini all'interno del veicolo.

I gateway saranno quindi costretti a continuare la loro evoluzione per poter soddisfare le modifiche architettoniche già previste che miglioreranno le specifiche tecniche delle IVN (larghezza di banda, latenza, prestazioni, sicurezza, ecc.).

Capitolo 6

Conclusioni

Dall'analisi prodotta risulta evidente come l'evoluzione dei settori dell'elettronica e delle telecomunicazioni abbia reso possibile uno sviluppo esponenziale delle tecnologie oggi comunemente incorporate nei veicoli, come i sistemi avanzati di assistenza alla guida. Requisiti estremamente restrittivi vengono oggi richiesti a tutti i produttori di applicazioni di questo tipo per garantire la sicurezza dei passeggeri coinvolti nel traffico urbano, il ché si traduce nella necessità di scambiare ed elaborare grandi moli di dati in frangenti di tempo sempre più piccoli.

Come risulta dallo studio condotto sul CAN bus, questo protocollo è stato di fondamentale importanza, soprattutto in ambito automotive, per permettere l'implementazione di applicazioni distribuite. Attraverso la comunicazione tra nodi di una rete, che avviene su un bus seriale robusto ed economico, è infatti possibile costruire complessi sistemi in cui ogni parte contribuisce in maniera propria al raggiungimento di un certo obiettivo. Oggi però, la quantità di informazioni che devono essere scambiate tra i nodi, nonché i tempi in cui ciò deve avvenire, rende insufficiente il data-rate massimo del CAN.

Nasce così la necessità di rendere eterogenea la rete onboard: non si ha più una sola linea di comunicazione CAN, ma ad essa vengono interfacciati sistemi LIN (per le applicazioni che non influenzano in nessun modo la sicurezza del veicolo) e FlexRay (per quelle che invece richiedono imponenti caratteristiche prestazionali). In questo modo si crea quindi un'architettura complessa, composta da più parti, in cui ognuna è dedicata al controllo e la gestione di attività che hanno requisiti comuni attraverso un protocollo ottimizzato proprio per quella funzione.

Data la larghezza di banda richiesta dalle moderne applicazioni e la loro ormai comune necessità di essere collegate in rete, risulta evidente quindi come Ethernet ed i sistemi IP-based rappresentino una soluzione perfetta, in grado di favorire sia l'integrazione sicura di più protocolli sullo stesso veicolo che lo sviluppo della connettività della vettura con il mondo circostante.

L'eterogeneità delle moderne IVN introduce così l'essenziale necessità di un dispositivo di rete in grado di permettere il dialogo tra sottoreti di tipo diverso. L'Automotive Gateway si presenta quindi come soluzione a questo problema: ottimizzata rispetto ai sistemi implementati nelle comuni reti locali, infatti, questa versione mira a soddisfare gli stringenti requisiti (in termini di sicurezza ed efficienza) del settore automotive ed agevolare lo sviluppo e la distribuzione dei veicoli connessi.

Personalmente ritengo di estremo interesse il coinvolgimento di tecnologie IP-based in un contesto veicolare, non solo per gli ovvi vantaggi in termini prestazionali che esse comportano, ma soprattutto per la vasta gamma di applicazioni innovative che possono così essere introdotte. Lo sviluppo di dispositivi ADAS e di sistemi per la guida autonoma, ad esempio, aumenta la necessità dei fornitori di questi servizi di possedere conoscenze avanzate in materia di programmazione, algoritmi, sicurezza e reti, dando così spazio in questo mondo, finora limitato a meccanici ed elettronici, anche agli informatici.

Bibliografia

[1] *Connected Car*, Wikipedia

[2] *Vehicle Bus*, Wikipedia

[3] *In-Vehicle Network*, IEEE Xplore

[4] *On-Board Diagnostics*, Wikipedia

[5] *Advanced Driver-Assistance Systems*, Wikipedia

[6] *Dedicated Short-Range Communications*, Wikipedia

[7] *Basic Safety Message*, US Department of Transportation

[8] *Intelligent Transportation System*, Wikipedia

[9] *Cooperative Vehicle Infrastructure System*, ReportLinker

[10] *Local Interconnect Network*, Wikipedia

[11] *Controller Area Network*, Wikipedia

[12] *FlexRay*, Wikipedia

[13] *Drive by wire*, Wikipedia

[14] *Automotive Ethernet*, IXIA

[15] *Controller Area Network (CAN)*, Wikipedia

[16] *Electromagnetic compatibility (EMC)*, Wikipedia

[17] *Electronic Control Unit(ECU)*, Wikipedia

[18] *RS-485*, Wikipedia

[19] *Robert Bosch GmbH*, Wikipedia

[20] *Society of Automotive Engineers (SAE)*, Wikipedia

[21] *International Organization for Standardization (ISO)*, Wikipedia

[22] *CANopen*, Wikipedia

[23] *Carrier Sense Multiple Access with Bitwise Arbitration (CSMA/BA)*, Wikipedia

[24] *Man-in-the-middle attack*, Wikipedia

[25] *Modello ISO/OSI*, Wikipedia

[26] *CAN Database Files*, National Instruments

[27] *Data Link Layer (DLL)*, Wikipedia

[28] *Logical Link Control (LLC)*, Wikipedia

[29] *Medium Access Control (MAC)*, Wikipedia

[30] *Media Access Unit (MAU)*, Wikipedia

[31] *Medium-Dependent Interface (MDI)*, Wikipedia

[32] *ISO 11898-1:2015*, ISO

[33] *ISO 11898-2:2016*, ISO

[34] *ISO 11898-3:2006*, ISO

[35] *ISO 11898-4:2004*, ISO

[36] *ISO 11898-5:2007*, ISO

[37] *ISO 11992-1:2003*, ISO

[38] *SAE J2411*, SAE

[39] *CAN Flexible Data-Rate (CAN FD)*, Wikipedia

[40] *Universal Measurement and Calibration Protocol (XCP)*, Wikipedia

[41] *LIDAR*, Wikipedia

[42] *IEEE 802.3*, Wikipedia

[43] *IEEE 802.1Q*, Wikipedia

[44] *1609.12-2019 - IEEE Standard for WAVE*, IEEE SA

[45] *Vehicular Ad-hoc Network (VANET)*, Wikipedia

[46] *Proxy Mobile IPv6*, Wikipedia

[47] *Radio Access Technology (RAT)*, Wikipedia

[48] *IEEE Vehicular Technology Society*, Wikipedia

[49] *DSRC/WAVE Technology*, Springer Link

[50] *ETSI EN 302 636-6-1*, ETSI

[51] *Communications Access for Land Mobiles (CALM)*, Wikipedia

[52] *Dynamic Host Configuration Protocol (DHCP)*, Wikipedia

[53] *Vehicular Address Configuration (CAV)*, UCLA

[54] *Mobile Ad-hoc Network (MANET)*, Wikipedia

[55] *DHCPv6*, Wikipedia

[56] *Car-To-Car Communication Consortium*, C2C

[57] *WiMAX*, Wikipedia

[58] *Simulation of Urban MObility (SUMO)*, Wikipedia

[59] *OpenFlow*, Wikipedia

[60] *Network Mobility (NEMO) Basic Support Protocol*, IETF

[61] *WPA2*, Wikipedia

[62] *Kerberos*, Wikipedia

[63] *"VIP-WAVE: On the Feasibility of IP Communications in 802.11p Vehicular Networks"*, IEEE Transactions on Intelligent Transportation Systems, March 2013

[64] *"IP Mobility Management for Vehicular Communication Networks: Challenges and Solutions"*, IEEE Communications Magazine, May 2011

[65] *"IPv6 Operation for WAVE - Wireless Access in Vehicular Environments"*, IEEE Vehicular Networking Conference, December 2010

[66] *"Mobile Internet Access in FleetNet"*, 13th Fachtagung Kommunikation in verteilten Systemen, February 2001

[67] *"A Layered Architecture for Vehicular Delay-Tolerant Networks"*, IEEE Symposium on Computers and Communications, July 2009

[68] *"Automatic IP Address Configuration in VANETs"*, ACM International Workshop on Vehicular Inter-Networking, September 2016

[69] *"Routing and Address Assignment using Lane/Position Information in a Vehicular Ad-hoc Network"*, IEEE Asia-Pacific Services Computing Conference, December 2008

[70] *"GeoSAC - Scalable Address Autoconfiguration for VANET Using Geographic Networking Concepts"*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, September 2008

[71] *"Experimental Evaluation for IPv6 over VANET Geographic Routing"*, IEEE International Wireless Communications and Mobile Computing Conference, June 2010

[72] *"Location-Aided Gateway Advertisement and Discovery Protocol for VANets"*, IEEE Transactions on Vehicular Technology, Vol. 59, No. 8, October 2010

[73] *"A Hybrid Centralized-Distributed Mobility Management for Supporting Highly Mobile Users"*, IEEE International Conference on Communications, June 2015

[74] *"A Hybrid Centralized-Distributed Mobility Management Architecture for Network Mobility"*, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2015

[75] *"SDN-based Distributed Mobility Management for 5G Networks"*, IEEE Wireless Communications and Networking Conference, April 2016

[76] *"Network Mobility Protocol for Vehicular Ad Hoc Networks"*, Wiley International Journal of Communication Systems, November 2014

[77] *"A Novel Mobility Management Scheme for Integration of Vehicular Ad Hoc Networks and Fixed IP Networks"*, Springer Mobile Networks and Applications, February 2010

[78] *"Securing Vehicular IPv6 Communications"*, IEEE Transactions on Dependable and Secure Computing, January 2016

[79] *"Providing Authentication and Access Control in Vehicular Network Environment"*, IFIP TC-11 International Information Security Conference, May 2006

[80] *Automotive Gateway*, NXP

[81] *Jitter*, Wikipedia

[82] *Diagnostic-over-IP (DoIP)*, AUTOSAR

[83] *Unified Diagnostic Services (UDS)*, Wikipedia

[84] *Over-The-Air (OTA)*, Wikipedia

[85] *NXP Semiconductors N.V.*, NXP

[86] *Hardware Security Module (HSM)*, Wikipedia

[87] *Interrupt Service Routine (ISR)*, Wikipedia

[88] *Routing table*, Wikipedia

[89] *Central Gateway, Bosch*, Bosch Mobility Solutions

[90] *Connectivity Control Unit (CCU)*, Bosch Mobility Solutions

[91] *Intrusion Detection System (IDS)*, Wikipedia

[92] *FEV Motorentechnik*, Wikipedia

[93] *Connected Vehicle Gateway (GATEWAY 5)*, FEV

[94] *Gateway Framework for In-Vehicle Networks Based on CAN, FlexRay, and Ethernet*, IEEE Xplore

[95] *Time-Division Multiple Access (TDMA)*, Wikipedia

[96] *Microcontroller Unit (MCU)*, Wikipedia

[97] *Protocol Data Unit (PDU)*, Wikipedia

[98] *OSEK/VDX*, Wikipedia

[99] *AUTomotive Open System ARchitecture (AUTOSAR)*, Wikipedia

[100] *Extensible Markup Language*, Wikipedia

[101] *Vector Informatik*, Wikipedia

[102] *CANoe*, Wikipedia

[103] *AUTOSAR Microcontroller Abstraction Layer (MCAL)*, embitel

[104] *MISRA C*, Wikipedia

[105] *CodeSonar*, Wikipedia

[106] *GrammaTech*, Wikipedia

Elenco delle figure