Towards Cloud-based Vehicular Cyber-Physical Systems

Kazi Masudul Alam

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

DOCTORATE IN PHILOSOPHY

in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada

June, 2017

ABSTRACT

We are living in the age of information technology, where we are fully occupied with the revolutionary innovations of the last few decades such as the Internet, mobile devices, wireless communications, social networks, wearables, cloud computing, etc. While these technologies have become integral part of our daily life, we are now anxiously waiting to embrace Internet-of-Things (IoT), intelligent digital assistants, driver-less cars, drone deliveries, virtual reality, and smart city applications. Recently, research community is demonstrating increasing interests about Cyber-Physical Systems (CPS) that resides in the cross-section of embedded systems, network communications, and scalable distributed infrastructures.

The main responsibility of a CPS is to collect sensory data about the physical world and to inform the computation module using communication technologies that processes the data, identifies important insights and notifies back using a feedback loop. These notifications can however be control commands to reconfigure the physical world. Such a setup is a useful method to deploy smart city applications. In this dissertation, we keep our focus onto the smart transport objective using vehicular CPS (VCPS) based systems organization. We have compiled this dissertation with our research contributions in this growing field of VCPS.

One of our key contributions in this field is an architecture reference model for the cloud-based CPS, C2PS, where we analytically describe the key properties of a CPS: computation, communication and control, while integrating cloud features to it. We have identified various types of computation and interaction modes of this paradigm as well as describe Bayesian network and fuzzy logic based smart connection to select a mode at any time.

It is considered that the true adoption of CPS is only possible through the deployment of the IoT systems. Thus, it is important to have IoT as a foundation in the CPS architectures. Our next contribution is to leverage existing Vehicular Adhoc Network (VANET) technologies and map them with the standard IoT-Architecture reference model to design the VCPS, Social Internet-of-Vehicles (SIoV). In this process, we have identified the social structures and system interactions among the subsystems involved in the SIoV. We also present a message structure to facilitate different types of SIoV interactions.

The ability of dynamic reconfiguration in a C2PS is very appealing. We capture this feature in the VCPS by designing a model-based reconfiguration scheme for the SIoV, where we measure the data workloads of distinct subsystems involved in various types of SIoV interactions. We further use these models to design dynamic adaptation schemes for the subsystems involved in VCPS interactions.

Our final contribution is an application development platform based on C2PS design technique that uses server-client based system communications. In this platform, server side is built using JAVA, client side uses Android, message communication uses JSON and every component has its own MySQL database to store the interactions. We use this platform to emulate and deploy SIoV related applications and scenarios. Such a platform is necessary to continue C2PS related research and developments in the laboratory environment.

ACKNOWLEDGMENTS

At first, I want to thank the almighty creator for His bounties and blessings into my life. I have been blessed with great family, exceptional mentors, amazing friends and superlative supports from the Canadian and the Bangladeshi Governments. Their continuous motivations always encouraged me to reach new milestones in my life. This Ph.D. dissertation is one of those greatest achievements.

I would like to express my sincere gratitude to my supervisor, Prof. Abdulmotaleb El Saddik, for his exemplary guidelines and visions that he sets forth in front of us everyday so that we can become better persons and good researchers with every new step. I thank him for his continuous supports, trust and guidance while pursuing this research work.

Special and sincere thanks to Dr. Mukesh Kumar Saini, Assistant Professor, IIT Ropar, India, for his assistance, friendship and continuous feedbacks, while the development of the thesis contributions.

I would also like to greatly thank my parents, Shahidul and Hosneara, for their inspiration, understanding, patience and endless prayers.

Very special appreciation to my dearest wife, Aysha and my lovely children Tahmeed, Sameeha and Sarah for their love, supports and sacrifices throughout this dissertation period. Also, thanks to my younger brother, Tauhidul, for keeping supports behind my back always.

I would like to thank Ontario Student Assistance Program (OSAP), Ontario Graduate Scholarship (OGS), University of Ottawa Financial Aid and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial supports throughout my doctoral program.

Finally, I would like to convey my appreciations to my colleagues Juan Sebastian Arteaga, Alex Sopena, Seyed Vahid Hosseinioun, Faisal Arafsha, Basim Hafidh, Mohammed Bin Hariz, Kafi Mahfuz and to my other colleagues at the MCRLab for their time and friendships throughout this research work. Thanks to my elderly friend Mr. Mohammad Tarique for his endless drive around in the Ottawa roads and philosophical coffee breaks. I would also like to thank Prof. Wail Gueaieb for maintaining a nice LaTeX template for the University of Ottawa Engineering thesis.

Dedicated to my extended family

Table of Contents

List of Tables

List of Figures

List of Acronyms

CPS	Cyber-Physical System
C2PS	Cloud-based Cyber-Physical Systems
VCPS	Vehicular Cyber-Physical System
M2M	Machine-to-Machine
IoT	Internet-of-Things
IoV	Internet-of-Vehicles
ITS	Intelligent Transport Systems
DFD	Data Flow Diagram
VANET	Vehicular Ad-hoc Network
DSRC	Dedicated Short-Range Communications
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2X	Vehicle-to-Others
SIoT	Social Internet of Things
SIoV	Social Internet of Vehicles
SenAS	Sensing-as-a-Service
ISP	Internet Service Providers
P2P	Peer-to-Peer
WSN	Wireless Sensors Networks
VSN	Vehicular Social Network
SaaS	Software-as-a-Service
PaaS	Platform-as-a-Service
IaaS	Infrastructure-as-a-Service
DAG	Directed Acyclic Graph
GPS	Global Positioning Systems
OBD II	On Board Diagnostics II
OBU	On Board Unit
RSU	Road Side Unit
HBU	Home Base Unit

Chapter 1

Introduction

1.1 Background

The urbanization of the eighties' and nineties' have contributed heavily to the increased embracement of city life over the rural ones. A United Nations report forecasts that by 2050 around 66% of the world population will be living a metropolis life that could add another 2.5 billion people to the urban centers. The growing technological advancements of the 21st century have already conceived the concept of Smart Cities. The definition of Smart City is rather ambiguous and has also been addressed in the literature as digital city, ubiquitous city, knowledge city, intelligent city, sustainable city, etc. Overall, a city can be defined 'smart' if it improves the quality of living using proper synergy of inhabitants' knowledge, traditional and modern communication infrastructures, information technology, efficient use of natural resources and participatory good governance [6][7][8].

The key stakeholders involved in Smart City projects are the city authorities, public and private sectors, utility service providers, academics and the residents. Different public and/or private organizations can provide city services such as water, energy and waste management; emergency and amusement services; public transport and traffic regulation, etc. If we integrate information technology and telecommunication services in these systems and the processes, we can ensure quality services in a Smart City. Technology will enable online data sensing and dissemination, which will ensure offline data analytics based active monitoring of overall services in a smart city [8].

In a Smart City, all the physical objects (i.e. *Things*) shall have embedded computing and communication capabilities so that they can sense the environment and cooperate with each other using wired or wireless communications to ensure high quality services for the users. These increasingly intelligent interconnections and interoperability often referred as Machine-to-Machine (M2M) interactions or the Internet-of-Things (IoT). Some of the important services in a Smart City are the Intelligent Transport Systems (ITS), Smart Water, Smart Energy, Smart Building and Waste Management, Culture and Tourism. From an efficient and productive city navigation perspective, ITS or Internet of Vehicles (IoV) is the most important domain for sustainable IoT adoption [9][10][8].

The number of vehicles have increased dramatically in recent times [11][12]. Almost all major cities experience heavy traffic during peak hours. An unfortunate accident or even a small road maintenance task can cause a huge traffic jam and further accidents. In the US alone, more than 16,000 crashes take place every day on highways [13]. Driver fatigue and lack of the early-warning system are responsible for these crashes [14]. Watchful suggestions from the surrounding vehicles could be vital in these cases to provide improved safety to the vehicle users. In a Smart City, an accident can automatically initiate the following services.

- vehicle sensors synchronization
- emergency medical services
- hospital arrangement
- traffic control
- towing service

- insurance provider notification
- police dispatch

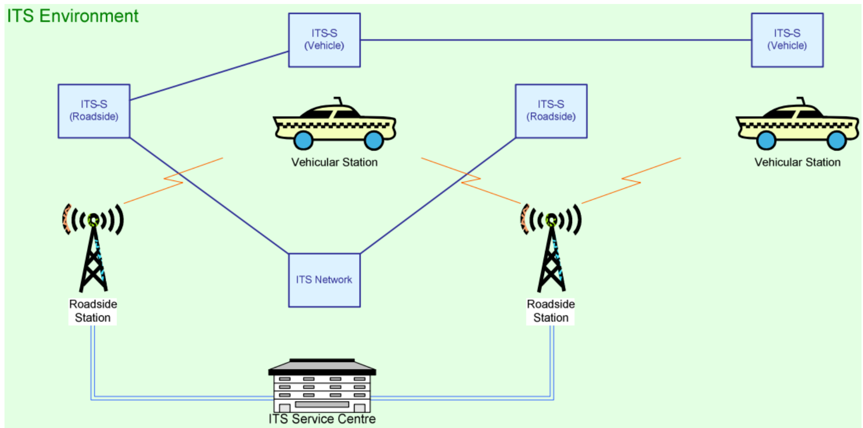


Figure 1.1: Simplified view of the ITS environment [15]

The ITS architecture guideline[1] of Canada identified 36 user services such as safety readiness, emergency notification and personal security, on-board safety and security monitoring, en-route transit information, traffic control, traveller services information (e.g. Local e-commerce), etc. Figure 1.2 shows the partial data flow diagram of the vehicle status monitoring process. This process suggests that on-board vehicular sensory data should be processed to find possible safety warnings and produce collision or crash avoidance guidelines.

State-of-the-art vehicles are increasingly equipped with cutting-edge sensors and different wireless communication capabilities[2] that can detect various in-vehicle and other surrounding events [16][17]. One of the advanced data dissemination approaches is to create vehicular ad-hoc networks (VANETs) with the neighboring vehicles using dedicated short-range communications (DSRC) technology, which is a two-way short-to-medium-range wireless communications capability with a high data transmission performance [18][19]. VANETs is a subject of on going research ([20, 21, 22, 23, 24]) for a while now, which is currently trending towards the Internet of Vehicles architecture [25, 26, 27].

[1]ITS architecture of Canada, http://tac-atc.ca/en/itsarchitecture, Accessed, December, 2015
[2]Tesla Model S and X, http://teslatap.com/undocumented/, Accessed, December, 2015

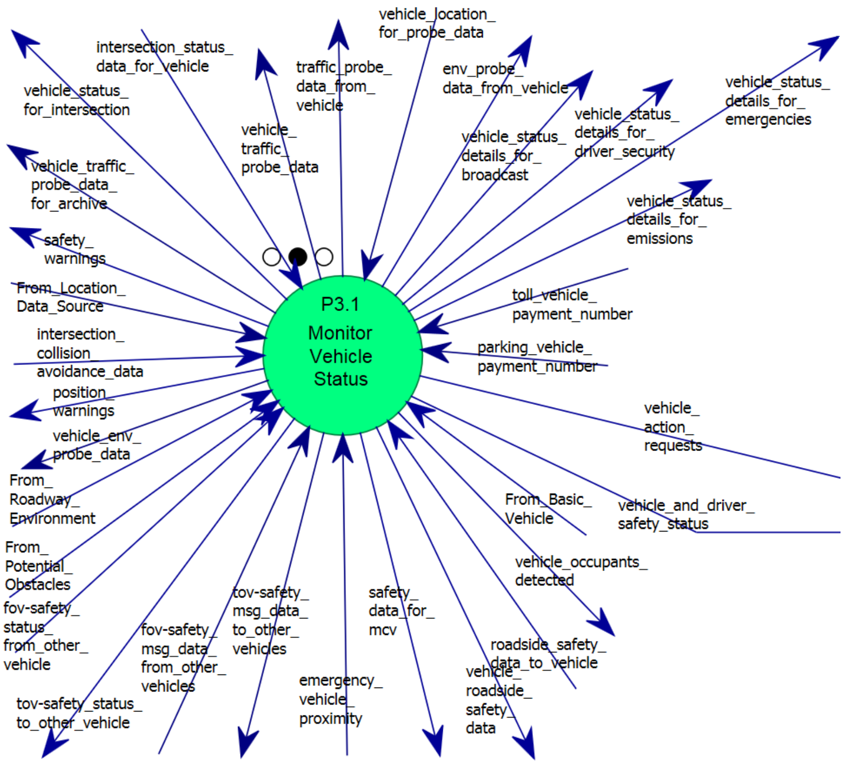


Figure 1.2: *Monitor Vehicle Status DFD* from the ITS architecture guideline of Transportation Association of Canada. The inward arrows indicate the data inputs to the process and outward arrows indicate data outputs from the process.

In the IoV model, every vehicle is considered as a physical hub for all the in-vehicle sensors and the surrounding mobile devices. The corresponding sensory data and the related analyses are the computation results that are used to invoke (i.e. control) smart vehicular decisions. All of this information is consumed from vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicle-to-others (V2X) (Figure 1.1) using multi-communication model. The inherent computation, communication and control aspects of the IoV systems fall in the Cyber-Physical System (CPS) category, where physical processes and software components are deeply intertwined to offer higher degree of automation, scalability and reconfiguration using multiple scales of networking [28][29]. The CPS aspects in the vehicular domain, vehicular cyber-physical system (VCPS) [30][31], is still in its infancy that requires further developments.

1.2 Motivations

The advancement in sensors and actuators development in the recent days improved their availability and affordability. As a result, invisible presence of versatile sensors, subsequent data acquisition using the computer networks and data analysis based control of the physical environment has become more realistic than ever. The properties of CPS solutions properly aligned in this phenomenon [32][5]. The authors in [32] proposed an architecture, named 5C, for the CPS to reach the goal of resilient, intelligent, and self-adaptable machines. In this architecture manufacturing systems support plug & play smart connections, provides smart analytics for the subsystems' health, instills cognition for decision making etc. Scalability of the 5C architecture can be improved by adopting cloud technology in the cyber, cognition and configuration levels. Cloud infrastructure can provide high-performance computing and vast storage facilities to the CPS.

Vehicular cyber-physical system (VCPS) [33][34] is a branch of CPS that can be of two types: intra-vehicle CPS and inter-vehicle CPS. In case of the intra-vehicle CPS, all the sensors inside the vehicles and other human factor mobile devices are used to infer local knowledge by applying sensory fusion. This process follows a tight integration of embedded systems communication in a limited networking capacity. Whereas, inter-vehicle CPS can extend the local knowledge of a vehicle with the collaborative information from other surrounding vehicles or friendly information sources (e.g. Infrastructure, People, IoT Sensors). Inter-vehicle CPS can be further extended by combining cloud infrastructures and social network features. Intra-vehicle CPS is useful for real time vehicular safety applications, whereas inter-vehicle CPS can provide optimized traffic and other comfort applications [5].

The underlying technology for the CPS related developments is the Ubiquitous Computing or Ubicomp. Ubicomp related researches introduced sensors in our ambient life that lately has initiated the revolution of IoT. Ning and Wang provided an architecture of future Internet of Things (IoT) using human neural network structure [35]. They defined a *Unit IoT* and combined various *Unit IoT*s to form the *Ubiquitous IoT*. Matthias et al. described a so called socio-technical network for IoT where every physical object is enabled with sensors to detect activity and later synchronizes the status using human readable short texts in Twitter[3] [36]. They present a proof-of-concept twittering plant application that shares moisture, and temperature information using twits. Atzori et al. introduced Social Internet of Things (SIoT) terminology that focuses on establishing and exploiting social relationships among *things* rather than their owners [37][38]. They identified different types of *things* relations based on location, co-work, ownership, etc. The *things* can crawl in their social networks to discover other *things* or *services* which can be exploited to build various IoT applications. Such characteristics, however, match with the online social network theme but in this case applied for the machine-to-machine (M2M) communication. Hence, we can address it as M2M social network.

A new model to describe the IoT is Sensing-as-a-service (SenAS) [39], where four conceptual layers are involved from the data provider up to the consumption process. In this model, *Sensors* are deployed to collect data about the environment and the sensor owners have the right to publish the sensor services. Here, *Sensor Publishers* (SP) play the broker role such as Internet Service Providers (ISP), where the sensors and their related services are published, and any service requests are forwarded through them. *Extended Service Providers* (ESP) offer the intelligent value-added services that analyses the consumer requests and search for the sensors and related services that would fit the payment model or the expected quality. *Consumers* are in the final layer of SenAS which ranges from individuals, private or public organizations, to the governments. Wide adoption of SenAS depends on the scalability of the system, which is easily available through the cloud infrastructure.

The technological intersection of the CPS, M2M social networks and the cloud tech-

[3]Twitter, https://twitter.com/

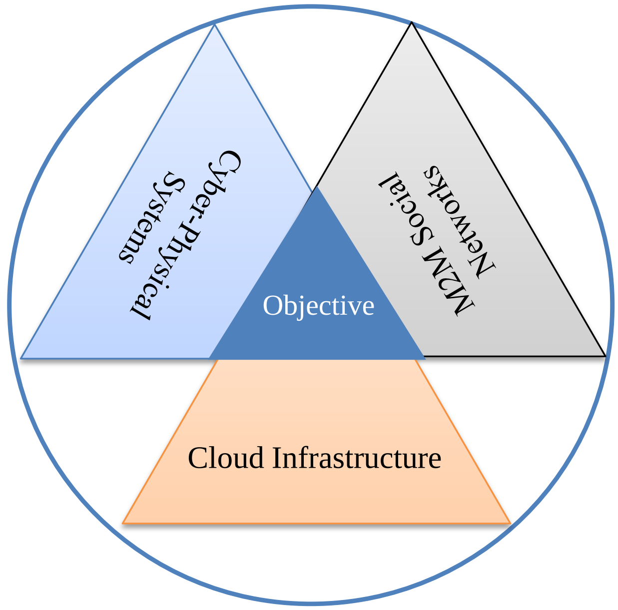


Figure 1.3: The technological intersection that is discussed in this dissertation.

nologies (Figure 1.3) can capture the required setup for an active Smart City, where physical environment can be sensed in real time and derived information can be meaningfully shared across different IoT domains to ensure efficient usage of resources to improve user experience and overall well being. The segregated cyber-physical layers of CPS allows independent evolution of both the physical and the cyber layer while keeping close integration among them. As a result, physical layer can provide real time sensory fusion and the cyber layer can extend that experience to support delay tolerant applications for the Smart City. Peer-to-peer social networking capability among the physical machines through the cyber layer ensures scalability across the physical networks which accomplishes cross domain IoT data sharing while making it easily navigable and privacy sensitive. Cloud technology provides the cyber layer with high performance infrastructure resources and data analytics capability, which can improve the CPS feedback control.

1.3 Objective

In this dissertation, we describe the technological intersection of the cyber-physical systems, cloud infrastructure and the M2M social networks from the vehicular interaction and services' perspective. This type of vehicular services model will advance current VANETs technology and will introduce more scalable vehicular CPS architecture, which can accommodate both real time and delay tolerant inter-vehicle or intra-vehicle services in the same platform. In order to accomplish this objective, we have to consult the guidelines from Internet-of-Things (IoT) implementation, which is considered as the foundation for effective CPS deployment. This approach will ensure that the VCPS architecture is open for integration with other domains of IoT, which is a key obstacle for smart city adoptions. In short, our main objective is to design and develop a cloud based vehicular cyber-physical system architecture and to build vehicle related applications on that platform.

1.4 Thesis Contributions

In this Thesis, we conceptualize the integration of cyber-physical systems, cloud infrastructure and the M2M social networks to facilitate the development of both real time and delay tolerant vehicular applications. In this process, we incorporate cloud based model with the usual cyber layer of the CPS architecture that enables it with scalable social network like machine-to-machine interactions among the vehicles, infrastructures and the users. Our model introduces interactions among the CPS subsystems beyond their physical communication reach by using peer-to-peer communications among the cloud hosted digital twins. As a result, various degrees of computing, communication and control features become available to all the applications. The overall Thesis contributions (Figure 1.4) can be summarized as:

- Conceptualization and analytical modeling of the computation, communication and

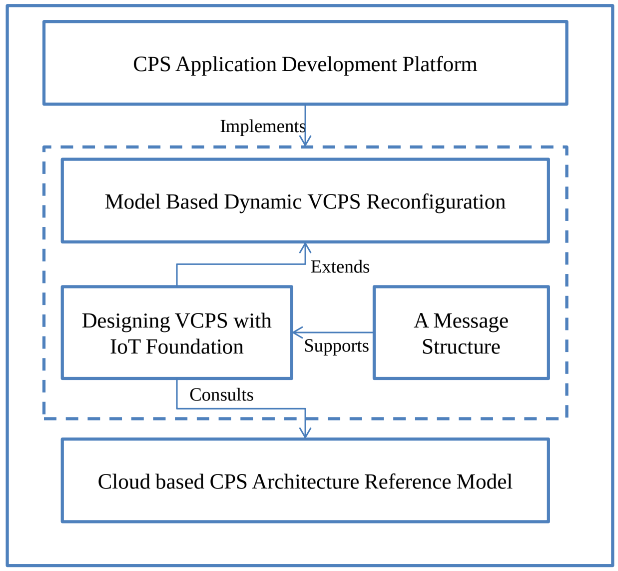


Figure 1.4: Summary of the Thesis contributions.

control properties of a cloud integrated cyber-physical system. We also provide the design details of a telematics based vehicle driving assistance application following the proposed cloud based CPS reference model.

- We leverage the existing VANETs technologies and map IoT Architecture Reference Model with it to introduce Social Internet-of-Vehicles i.e. Social IoV (SIoV). SIoV exploits social network like characteristics to describe the M2M relationships among vehicular CPS subsystems. We identified the social structures and interactions among VCPS subsystems and provide their architectural guidelines.

- We also designed and developed a communication message structure for the VCPS that hosts all the social structures and interactions among the subsystems. In this case, interaction data are stored in various levels of VCPS communications and can provide real time, delay tolerant and mining application services.

- We further designed and developed analytical models to estimate the data workloads of the individual subsystems involved in the VCPS interactions, which provides required tools to dynamically reconfigure the VCPS settings.

- Finally, we designed and developed an extendable cloud based CPS application development platform that was used to emulate VCPS interactions and for building safety, delay tolerant vehicular applications.

1.5 Scholarly Articles

Followings are the list of journal articles and conference papers that have already been published or in the process of publication in reputed peer-reviewed journals or conferences.

Journal articles:

- **Alam, Kazi Masudul**,, and Abdulmotaleb El Saddik. "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems." IEEE Access, 5 (2017): 2050-2062.
- **Alam, Kazi Masudul**, Mukesh Saini, and Abdulmotaleb El Saddik. "Toward social internet of vehicles: Concept, architecture, and applications." IEEE Access, 3 (2015): 343-357.
- **Alam, Kazi Masudul**, Mukesh Saini, and Abdulmotaleb El Saddik. "Workload Model Based Dynamic Adaptation of Social Internet of Vehicles." Sensors, 15.9 (2015): 23262-23285.
- Mukesh Saini, **Kazi Masudul Alam**, Haolin Guo, Abdulhameed Alelaiwi, and Abdulmotaleb El Saddik. "InCloud: a cloud-based middleware for vehicular infotainment systems." Multimedia Tools and Applications (2016): 1-29.

Conference Papers:

- **Alam, Kazi Masudul**, Mohammed Bin Hariz, Seyed Vahid Hosseinioun, Mukesh Saini, and Abdulmotaleb El Saddik. "MUDVA: A multi-sensory dataset for the vehicular CPS applications." In Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop on, pp. 1-6. IEEE, 2016.

- **Alam, Kazi Masudul**, Alex Sopena, and Abdulmotaleb El Saddik. "Design and Development of a Cloud Based Cyber-Physical Architecture for the Internet-of-Things." 2015 IEEE International Symposium on Multimedia (ISM). IEEE, 2015.

- **Alam, Kazi Masudul**, Mukesh Saini, Dewan T. Ahmed, and Abdulmotaleb El Saddik. "VeDi: A vehicular crowd-sourced video social network for VANETs." In Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on, pp. 738-745. IEEE, 2014.

- **Alam, Kazi Masudul**, Mukesh Saini, and Abdulmotaleb El Saddik. "tNote: A social network of vehicles under internet of things." In Internet of VehiclesTechnologies and Services, pp. 227-236. Springer International Publishing, 2014.

1.6 Thesis Organization

The rest of the dissertation is organized as follows:

- Chapter 2 presents required details about the key topics related to the Thesis content such as vehicular ad-hoc networks, cyber-physical systems, internet-of-things in the background section. The related works section discusses various state-of-the-art research works that closely relates to our contribution of cloud based vehicular CPS.

- Chapter 3 introduces the cloud based cyber-physical system (C2PS) architecture reference model that analytically describes the key characteristics of a CPS, computation, communication and control. We also explain an example of the C2PS, vehicular CPS (VCPS), in this chapter.

- Chapter 4 introduces the details of VCPS, Social Internet-of-Vehicles (SIoV), and discusses the social structures and interactions of VCPS subsystems. While designing the SIoV, we leverage the existing VANETs technologies and map them with the standard IoT Architecture Reference Model to make it more adoptable. A communication message structure, *tNote*, that also works as a data structure at various levels of VCPS interactions is introduced in this chapter.

- Chapter 5 presents the analytical models that have been designed to estimate the data workloads of individual VCPS subsystems. We also present a few adaptive algorithms that use these models to dynamically configure the VCPS settings for different vehicular scenarios.

- Chapter 6 provides the details about how we develop an extendable C2PS application development platform using JAVA, Android, MySQL, and JSON technologies. We discuss how we use the platform to emulate SIoV scenarios and also provide detailed experimental analysis related to the SIoV models. A telematics based application that provides assistance in vehicle driving is also described in this chapter. Additionally, we present an attempt to collect multi-sensory VCPS data as well as show how we use the platform and the dataset to build safety and delay tolerant applications.

- Chapter 7 presents the concluding remarks about the dissertation and discusses about the possible future works on the proposed topic.

Chapter 2

Background and Related Works

2.1 Background

In this section, we present the key background topics related to the dissertation. For this purpose, we present the fundamental details of Cloud computing, Vehicular Ad-hoc Networks (VANETs), Internet-of-Things (IoT), and Cyber-Physical Systems (CPS) that are required to understand the state-of-the-art works and subsequently the thesis contributions. VANETs describe the data sharing aspects of vehicles, infrastructures, and people while on the roadway. Whereas, IoT discusses about the techniques, how a plethora of smart things can connect in the Internet and provide effective services to improve the quality of our living. On the other hand, CPS is considered as the next generation of IoT evolution, when computation, communication and control are distributed, and smart decisions taken by the higher capacity systems will be fed back to the low-level physical systems to enable higher degree of control and reconfigurations. Cloud computing provides storage, networking, and computing services for all these systems to improve overall accessibility and scalability.

2.1.1 Cloud Computing

Cloud computing is a successful new trend, where the IT infrastructures from the organizations or the individuals are moved to a central computing, storage, and network service provider. This is an ondemand service model, where individuals and corporations are billed usually in pay-as-you-go model. People can buy a required amount of virtual computing resources from the cloud service providers, using simple to use web interfaces. Clouds reduce the cost of managing IT infrastructures and handover the responsibilities to expert providers. In theory infinite computing, storage services can be provided by the top cloud providers such as Microsoft Azure, Amazon Web Services, Google App Engine, etc. [40].

Cloud services can be usually of two types: private cloud and public cloud. In case of *private cloud*, all the cloud technologies are hosted inside an organization's infrastructure isolated from the public using firewalls, without bandwidth restrictions, and with required security or legal obligations about the business data. Whereas, *public cloud* is accessible anywhere through the Internet and websites or the data from different customers are stored inside the same infrastructure of a third party cloud provider. Public cloud usually follows pay-as-you-go model, where unused resources will be relinquished over time. *Hybrid cloud* is another approach that connects an organizational private cloud with a public cloud service. Some cloud providerse also offer virtual private network (VPN) based secured private cloud inside a public cloud, called *virtual private clouds* [41][40].

Cloud services can be divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) (Figure 2.1) [41][40].

- **Infrastructure-as-a-Service (IaaS):** Infrastructures such as computing power, storage capacity and networking are made accessible to the clients through easy to access web interfaces. In this model, users can easily increase or decrease resources based upon their cost and demand. A cloud service provider allows the users to create virtual machines that act like an independent machine in this model. Applications

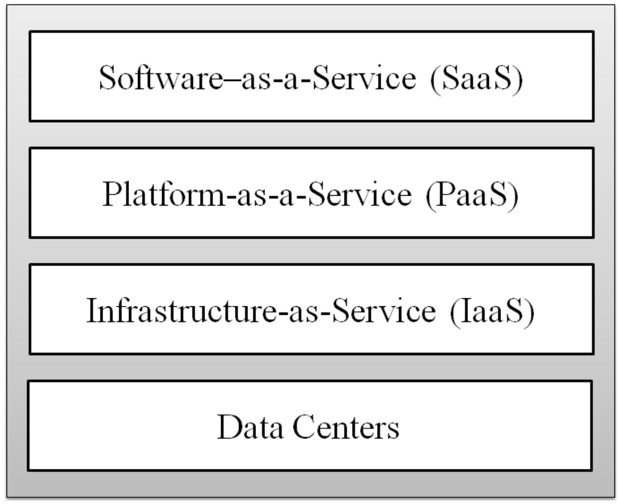


Figure 2.1: Layered cloud computing architecture, where data centers provide the infrastructures to work with different platforms and software setup. Each of this layer can provide a type of service to a customer.

are deployed in the virtual machines. Amazon EC2 is an example of IaaS.

- **Platform-as-a-Service (PaaS):** PaaS offers a new model of software development and deployment, where the cloud provides all the required systems, environments and tools required for software development, testing and deployment. This can help reduce the cost of IT operations in an organization. Microsoft Azure and Google Apps are examples of PaaS.

- **Software-as-a-Service (SaaS):** In the SaaS model, software is delivered as a service to many customers. Common resources and a single instance of application code or database provides support to multiple tenants in the SaaS. A SaaS provider manages hosted applications from the customers and might offer integration of third party hosted applications with the customers' application as well. Salesforce CRM system is a good example of SaaS.

2.1.2 Vehicular Ad-hoc Networks

The growing technological advancements in the field of information technology have made Smart Cities a thing of near future. In a Smart City, all objects would have embedded processors and capability to communicate with each other through wired or wireless connections. Vehicles (i.e. Intelligent Transport System) are one of the key elements of the Smart City. As the number of vehicles have increased dramatically in recent years, it is causing jams in the high speed roads during the peak hours. A small road maintenance task or accident can result in huge traffic jam or further accidents. Possible solutions to these problems are offered by the Vehicular Ad-hoc Networks (VANETs).

VANET enables vehicles with the capabilities to form ad-hoc networks and communicate with each other using wireless communication technologies (Figure 2.2). These vehicle-to-vehicle communications are termed as V2V communication. A vehicle can also be in communication with roadside static infrastructures such as traffic lights, base stations. These kinds of communications are grouped as vehicle-to-infrastructure (V2I) communications. Vehicles can also communicate directly with pedestrians or other Internet-of-Things (IoT) objects. We can group them as V2X communication.

Every vehicle is equipped with an onboard computing and communication unit called On-Board Units (OBU). Whereas, the roadside infrastructures come with Roadside Unit (RSU). An RSU has multiple network interfaces to communicate simultaneously with the vehicles, other RSUs and the base station. V2V communication is single hop, when two vehicles are in their communication range. Otherwise inside a vehicular platoon data is transferred through multi-hop communication. VANETs share some similarities with the Mobile Ad-hoc Networks (MANETs), but in VANETs topology changes quickly because of higher node speed. As a result, connection lives between two VANET nodes are short and signal quality is poor because of radio obstacles [3].

In the VANETs, the V2V and V2I, communications occur through the wireless channels.

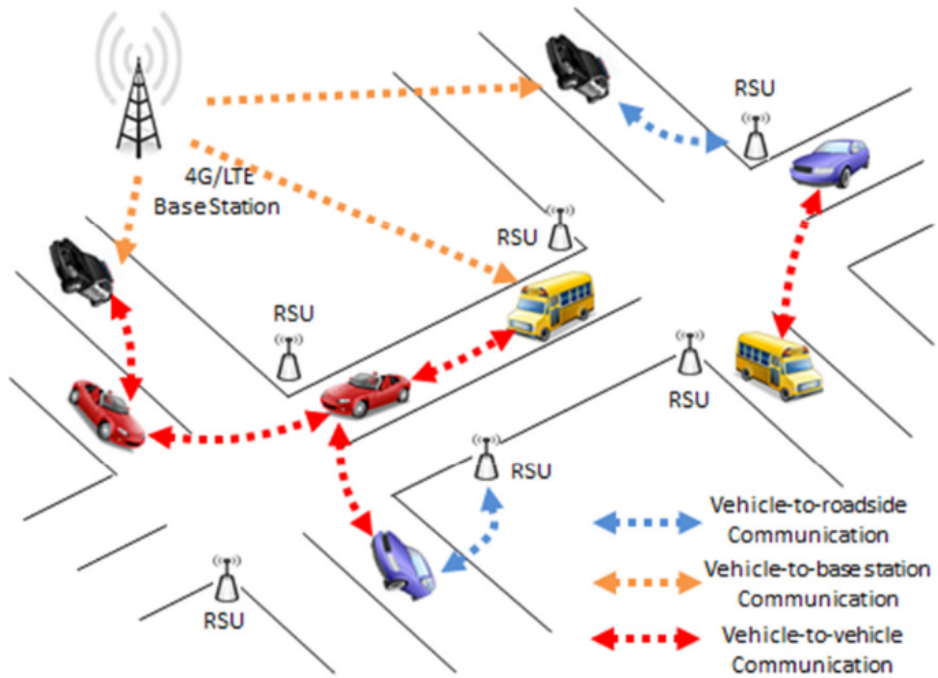


Figure 2.2: VANET components and communications [1]

Authorities from different regions such as North America, Europe, and Japan have allocated different frequency bands for the ITS applications. For example, Federal Communication Commission (FCC) of United States has allocated 75MHz bandwidth in the 5.9GHz band, where 5 MHz is reserved as the guard band and out of seven 10-MHz channels, 1 control channel (CCH) is used for high-priority short messages and other 6 channels are used as service channels (SCHs) (Figure 2.3) [2]. Here, 3 channels are assigned for safety related applications and the rest of the 4 channels can be used for non-safety related applications [3]. None of the contemporary wireless communication technologies such as Bluetooth, Wi-Fi, UTMS, and LTE are suitable to tackle versatile mobility, speed requirements of the VANETs. As a result a new communication stack WAVE (Wireless Access in Vehicular Environments) has been devised. Cellular networks such as UTMS and LTE are suitable to share data with the base stations, whereas WAVE is used for Dedicated Short Range Communication (DSRC). Safety applications are latency sensitive and require 10 messages per second. On the other hand, efficiency applications have a few seconds of delay tolerance.

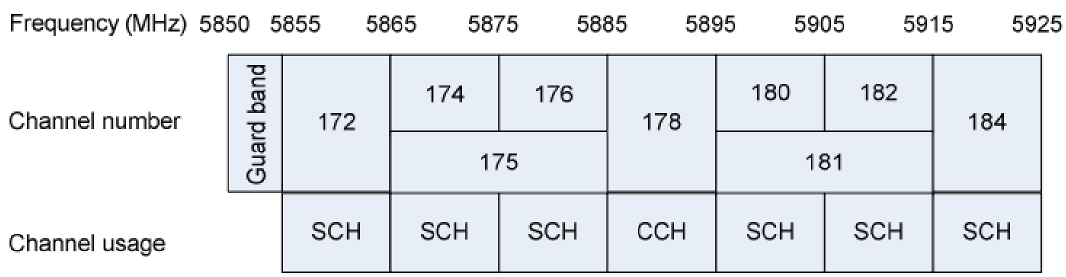


Figure 2.3: The DSRC Frequency Allocation in US [2]

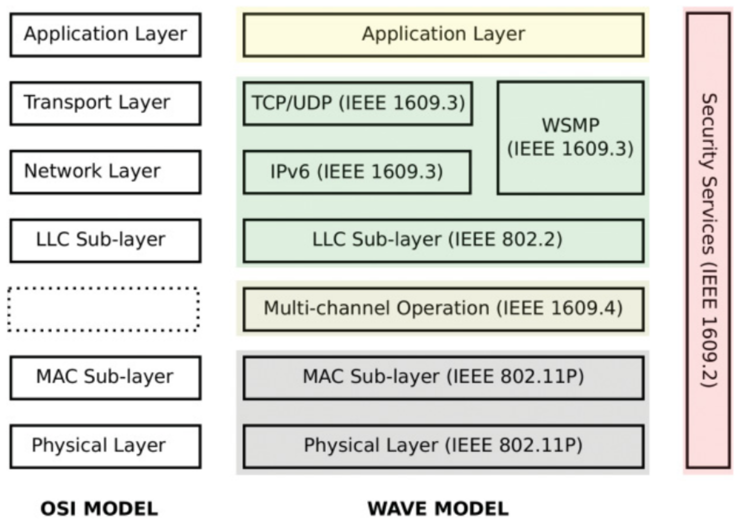


Figure 2.4: WAVE communication stack is partly based on OSI model to tackle VANET problems. WAVE standard of every layer are presented inside the parantheses [3].

Comfort and entertainment applications are not restricted to latency [3].

2.1.3 Internet-of-Things

Internet-of-Things (IoT) is a world wide network of interconnected smart objects, where every smart object has its own unique ID and is able to communicate with any other unique object using either wired or wireless communication methodology. Atzori et al. defined IoT paradigm as a convergence of three different visions (Figure 2.5): *Things oriented*, where every object is addressed uniquely; *Internet oriented*, where smart objects can be easily accessed through lightweight IP protocols, and *Semantic oriented*, where knowledge organization, search and reasoning are possible [4].

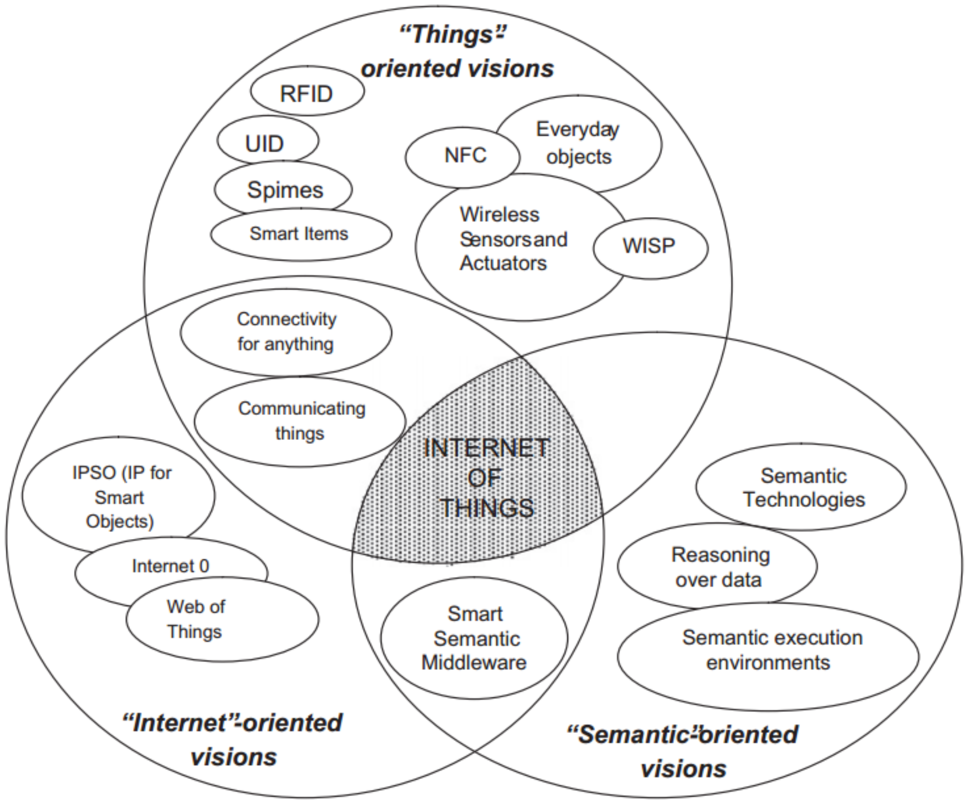


Figure 2.5: "Internet-of-Things" paradigm as a result of convergence from different visions [4].

Gubbi et al. described three IoT components to ensure seamless ubicomp: hardware, middleware and presentation. Radio Frequency Identification (RFID) technology offers a breakthrough in IoT hardware identification, which can be both active (battery based, initiate communication) or passive (no battery, one way identification). Low power miniature devices are useful for data collection in Wireless Sensor Networks (WSN). All the elements that are already connected to the Internet should be easily accessible either through the IPv4 or IPv6. Uniform Resource Name (URN) offers a better solution to uniquely address any IoT element. Lightweight IPv6 might be required to address home appliances. It is difficult to update existing WSN communication stacks. As a result, URN based subnet can be an effective solution to access low level sensors. IoT objects continuously generate useful information that can be stored in the scalable cloud infrastructures. We also need 2D/3D data visualization methods for IoT data. Effective deployment of IoT applications can ensure a smart environment of smart transport, smart city, smart water, smart agriculture, smart retail, and smart home to improve the Quality of Life (QoL) of users [42].

2.1.4 Cyber-Physical Systems

The objective of Cyber-Physical System (CPS) is to integrate computation, communication and control (Figure 2.6) in both the physical and cyber layer, so that the physical world can be actively monitored and modified by the cyber world and vice versa. In this complex system, physical processes produce observations about the real world and cyber resources compute, process and analyze the observations to reach necessary conclusions. These conclusions might trigger control actions initiated by the cyber resources to modify the physical world. The objective of these changes is to reach a common goal for different sub-systems involved in a CPS. Contemporary researchers have defined CPSs in different ways.

Lee first defined CPS as integration of computation and physical processes, where physical processes are monitored and controlled by embedded computers and networks. Physical processes affect computations using feedback loops and vice versa [43]. Rajkumar et al. later described CPS as physical and engineered systems, whose operations are monitored, coordinated, controlled and integrated by a computing and communication core [44]. Shi et al. described CPS as abstractions of physical process, software and communication that provides modeling, design and analysis techniques for the integrated whole [45]. NSF defined CPS as the systems, where physical and software components are deeply intertwined and operating in different spatial and temporal scales with myriad of contextual behaviors [46]. Khaitan et al. described CPS as a complex and heterogeneous *System of Systems* that requires proper regulation and careful codesign of the overall architecture [29].

In [47] authors describe CPS as an integration of physical systems with the cyber sys-

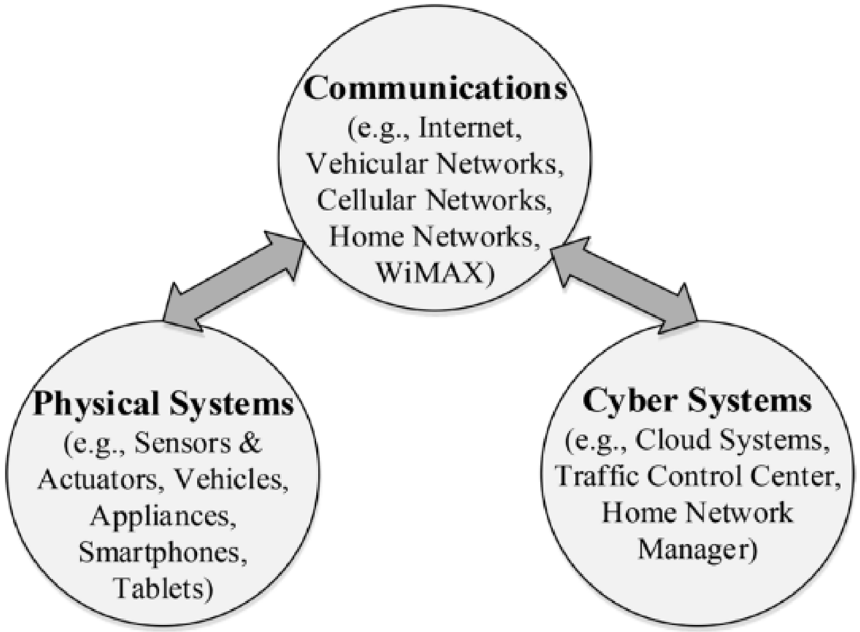


Figure 2.6: Cyber-physical systems (CPSs) as described in [47]

tems via communications. Effective policies are required to adapt to these time-varying physical and cyber context. A CPS can adhere to a list of requirements: 1) cyber capability in the physical components, 2) higher degree of automation, 3) multi-layer networking, 4) reconfiguration capacity, 4) integration in temporal and spatial scales [29]. In a CPS safety and reliability requirements are higher than the general purpose computing. The economic and societal benefits of CPS applications are huge. The types of applications CPS can support are networked intelligent vehicles, connected process control, intelligent transport systems, smart grids, distributed defense infrastructures, smart buildings, connected homes, water resources, etc. [43].

Vehicular CPS is a subset of CPS where vehicles and other roadside infrastructures play together to ensure safe and entertaining travel in the road networks. A vehicular CPS can be of two types: intra VCPS and inter VCPS. In case of intra VCPS all the vehicular sensors inside/outside of a vehicle are involved in collecting vehicular events and driver behavior information that can be useful to make intelligent decisions by a smart vehicle. On the other hand, inter VCPS ensures that the sensory information gathered

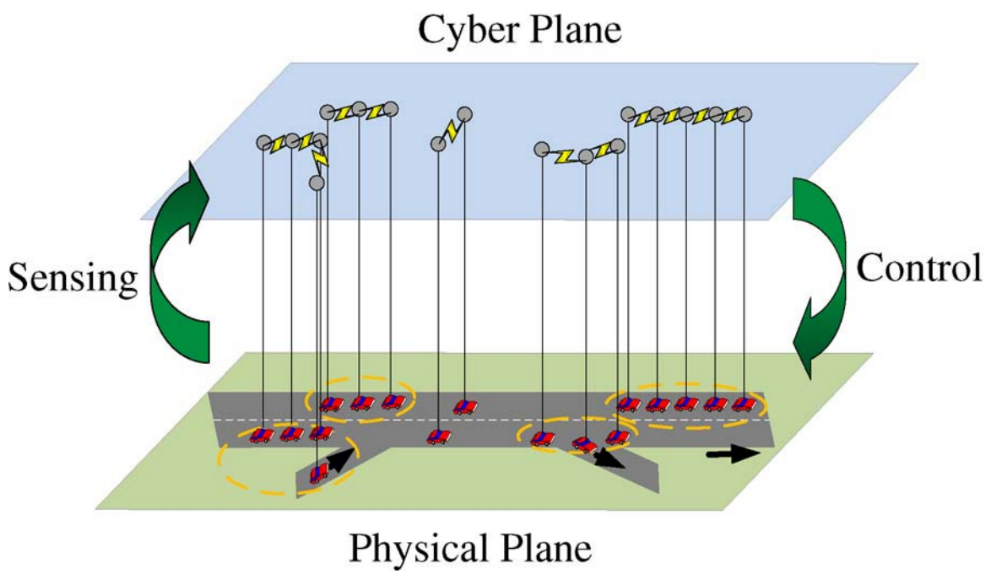


Figure 2.7: An illustration of a vehicular cyber-physical system [5]

by one vehicle is shared with its neighboring vehicles or with the roadside infrastructures or central data center using wireless communication techniques. Vehicles running in the road networks form platoons with and around their neighboring vehicles based on possible communication range (Figure 2.7).

2.2 Related Works

In this section, we discuss the closely related state-of-the-art works that we based our research on. We categorically divide this section according to the thesis contributions. At first, we present the state-of-the-art concepts related to the cyber-physical systems which lead to the proposal of cloud based CPS architecture reference model. Later, we describe the vehicular CPS related works, and finally the works related to the VCPS data workload modeling.

2.2.1 Cyber-Physical Systems

Guinard et al. [48] discussed how *Web-of-Things* can share their functionality interfaces using human social network infrastructures such as Facebook, Linkedin, Twitter, etc. In their system every object that wants to share its functionality to the web either has a built-in embedded web server, or proxy smart gateways (e.g. RFID tag based devices). The *Smart-Things* of an individual person share their web crawlable public interfaces with the owner's groups and friends through a social network. Friends and family get notifications about the shared smart things through the social network APIs. Operations on the shared things can be done through the RESTful `PUT, POST, GET,` etc. actions.

Smart-Its Friends [49] looked into how qualitative wireless connections can be established between *smart-artifacts*. In this system, every smart object is consist of two units: 1) data acquisition and generic feature extraction are managed by the sensor unit; 2) application specific processing, device control and communication with other smart-its compliant devices are handled by the core unit. Their system introduces context proximity based match making and respective connections. A possible application of the system is to monitor the presence of children in close proximity of their parents.

Ning and Wang provided an architecture of Future Internet of Things (IoT) using human neural network structure [35]. They defined *Unit IoT* as the man-like nervous (MLN) model that has three parts: brain (management and centralized data center: M&DC), spinal cord (distributed control nodes), and a network of nerves (IoT network and sensors). A combination of various *Unit IoT*s form the *Ubiquitous IoT* i.e. the global IoT. Global IoT includes industrial IoT, regional IoT, and national IoT. The overall global IoT is hierarchically structured and connected in a socially organized framework so that specific authority can control a small domain of IoT.

Matthias et al. describe a so called socio-technical network for IoT where every physical object is enabled with sensors to detect activity and later synchronizes the status using human readable short texts in Twitter [36]. Here, Twitter is a medium of communication among the things and the humans. Every thing or human both publish and subscribe to twitter feed of a thing or human to exchange information among them. They present a proof-of-concept twittering plant application which shares moisture, and temperature information in the twitter. In the winter time, light composition can be modified to suit the environment following the twitter message. This procedure is called perception-cognition-action loop.

Atzori et al. have introduced Social Internet of Things (SIoT) terminology and focuses on establishing and exploiting social relationships among *things* rather than their owners [37][38]. They have identified different types of *things* relations based on location, co-work, ownership, and social relationships. In the SIoT, a new *thing* is first registered on the system; later the available *services* of the smart thing are explored by other interested things. SIoT things can establish various relationships dictated by the owners or through matching the things' profiles. Once a service is requested by an application agent, related services searching and subsequent services composition are completed based on trustworthiness before the final information delivery. SIoT organizes the members in four classes based on their computational, communication and mobility properties.

Lee et al. [32] proposed a 5-level CPS architecture for industry 4.0 based manufacturing systems (Figure 2.8), which supports plug & play smart connection; provides smart analytics for subsystem health; enables twin model for components and machines; instills cognition for decision making and self-configuration for resilience. A CPS consists of two functions: 1) advanced connectivity to collect real time data about the physical world, 2) intelligent data management, analytics and computation in the cyber space. Tether-free and seamless connections are important for smart connections. Again, cyber space works as the central hub for data collection and processing. Cognition is achieved thorough the analysis of gathered data that leads to feedbacks as configuration adjustments.

Barthels et al. presented an intra CPS architecture [50] to manage power in automotive

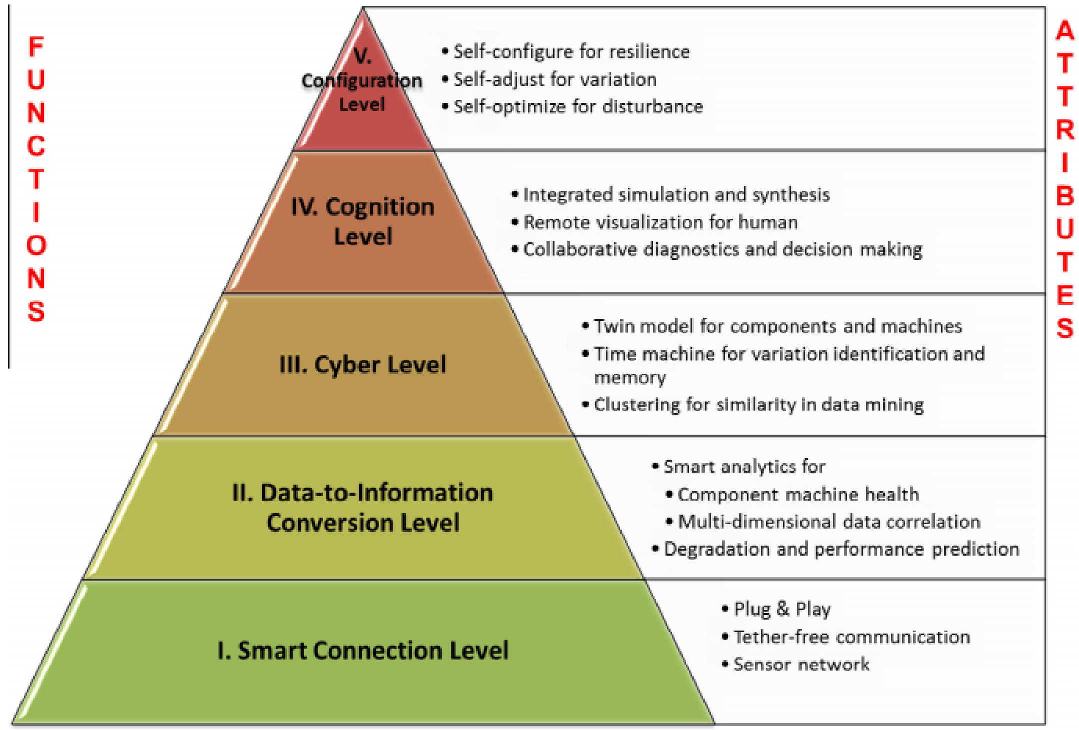


Figure 2.8: 5C architecture proposed for CPS [32]

systems. They represent the machine in functional state sequences, where physical input sequences transduce into different power management plan sequences. They used Moore's machine to represent power management subsystem, where a power management module is a transducing finite state, physical inputs trigger functional state transition and output functions are represented as power management plan. In [51] authors present a multi-tier architecture to integrate CPS and cloud computing to offer scalable control algorithms in the cloud and easier third party data source integration. In this architecture, embedded tiers are very lightweight and send commands, sensory value to the cloud tiers. Cloud tiers execute the control algorithms and send back results to the embedded tier. Their client tier serves as the human machine interface that can be either PC, tablet, smartphone or a web browser. They applied their architecture to improve IT and control aspects in the field of renewable energies (i.e. solar energy). Functional model based CPS design methodology is presented in [52], where authors use functional models for high level abstraction of multidisciplinary systems.

2.2.2 Vehicular Cyber-Physical Systems

Smaldone et al. introduced Vehicular Social Network (VSN)[53] at first in RoadSpeak [54], where they used the vehicular network for human socialization from entertainment, utility, and emergency messaging perspective. They described RoadSpeak voice chat application, where vehicle users can dynamically form voice chat groups (VCG) based on the location, time, route, and interests to engage in interactions. Every VCG is consist of the triplet {`time, location, interests`} and new users join a group based on user profile, location and time period matching.

Hu et al. proposed a service oriented architecture for the VSN named VSSA [55]. VSSA divides the system in application layer and service layer. In the service layer, they present *Aframe* service for easier collboration in the mobile ad-hoc networks, social services to integrate the system with popular social networks, and context services to collect contexts and derive understanding using inferencing. The application layer provides interfaces for traffic congestion detection, map route suggestions, etc. The proposed system does not clarify how VSSA can be integrated in the IoT scenario, the types of vehicular interactions or their inter-relationships, and the communication messaging structure.

Autonomous vehicular clouds (AVC) was first discussed in [56], where authors describe how ubiquitous internet presence in vehicles and immense opportunities of cloud computing resources can help to form the AVC. In this model, untapped computing, communication resources of current and future vehicles will provide necessary coordination and dynamic resources allocation opportunities to the authorized users. AVCs can provide short lived public Internet services to vehicles on the go or long time private services to a limited number of users such as a fleet of FedEx. The types of applications AVCs can support ranges from on-road safety, and traffic management to asset management.

Hu et al. also introduced *Social Drive* [57] system, which fuses onboard diagnostic (OBD) based vehicle information with the traditional social networks to present more user friendly feedback to the drivers. The system shares interesting information about a vehicle driver such as the total number of hard accelerations, total travel distance, usual driving routes, etc. that can be useful to ensure improved fuel economy. They developed an user-friendly Android mobile application to collect these data which can be accessed through the cloud based social web platforms as well. Though *Social Drive* is an interesting step in the overall VCPS direction, it does not provide vehicle-to-vehicle or vehicle-to-infrastructure interactions and also does not specify the required feedback control from cloud to the sensor layer.

Abid et al. proposed V-Cloud [30] system for vehicular communications, which is composed of three layers: in-car VCPS, vehicle-to-vehicle (V2V) network, and vehicle-to-infrastructure (V2I) network. In-car layer represents all types of sensors: vehicle's internal physical sensors and smartphone based sensors. Vehicles form V2V clusters on the highway, where each V2V cluster represents a cloud computing node. V2I provides the means to get connected with the central cloud computing infrastructure. These cloud infrastructures can be either public cloud such as Microsoft Azure, Amazon Web Services, etc. or built privately by traffic management authorities. This system overall ignores the inter-VCPS aspects required for successful adoption of the VCPS, by only focusing on the intra-VCPS aspects that is the in-car modules. Furthermore, the aspects of cyber-physical feedback control, which essentially relates it to the machine-to-machine (M2M) social networks are not discussed in this paper.

Hussain et al. [58] first introduced an architecture merging VANET and the cloud computing. In their system, they prefer SaaS and IaaS for suitable adaptation of the VANET cloud system. They provide a comprehensive VANET cloud taxonomy describing three types of VANET clouds: vehicular clouds, VANET using clouds (VuC) and hybrid clouds. Vehicular clouds can be two types, static and dynamic; static cloud is composed of stationary vehicles in a common location agreeing to form a cloud; and dynamic cloud is formed in real time based on an event or situation. In case of VuC, vehicles use cloud

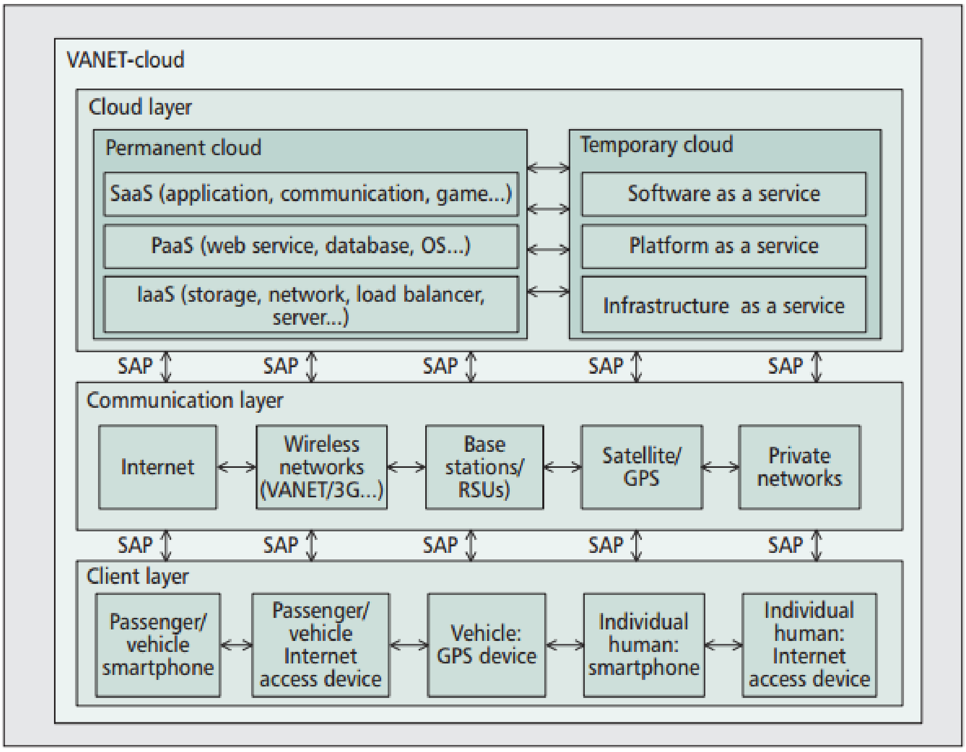


Figure 2.9: VANET-Cloud architecture proposed in [34]

services while moving and RSUs act as gateways to the central clouds. Hybrid cloud is a combination of both, where moving vehicle can rent its services as well as consume services from central cloud. Integration to the internet-of-things or feedback control for an active VCPS are not covered in this architecture.

In [34] authors present a general purpose VANET-Cloud system that uses both stationary and mobile entities of VANET by including onboard computers as an extension to the conventional cloud infrastructure. They divide the system in three layers: client, communication and cloud layers. End users reside in the client layer and access the services provided by the cloud layer through the intermediate communication layer. VANET-Cloud is composed of two sub-models: permanent cloud consists of the usual data center (i.e. virtual machines, storage, networking) and the temporary cloud consists of the mobile entities (i.e. vehicles, passenger devices, etc.). All the three types of cloud services Iaas, PaaS and SaaS are available for the users. Permanent and temporary VANET-Cloud are connected using necessary networking technologies and access the resources of each other. In this system, authors mainly focus on the computing and data storage aspects of a VANET-Cloud but their internal interactions, or feedback control from one layer to the other has not been discussed.

In [31], the authors propose an IoT based vehicular cloud platform that supports both V2V and V2I communications through the integration of sensors and mobile phones with middleware, and cloud computing technologies. The system provide both real time and on-demand services through the traditional and temporary cloud infrastructures. The conventional cloud is composed of virtualized computers, and the temporary cloud is composed of on the fly vehicular clouds. They use SOA based access points to integrate different layers of the architecture. They also present vehicular cloud services such as intelligent parking cloud service and vehicular mining data service as PaaS applications for the proposed system. Though the proposal is very promising but the integration of vehicles to the IoT architecture should follow some standard reference model to enable other IoT domain integrations. Moreover, in autonomous interaction scenario, the role of sensor cloud and action feedback model to the physical actuators need to be defined.

In [33], the authors present hierarchical VCPS architecture, VCMIA, where the *Micro* layer is responsible for vehicular sensory data collection to ensure safety, the *Meso* layer represents VANET based vehicular cluster to share safety and infotainment information, and the *Macro* layer represents the cloud data center that offers various traffic related services. Vehicles and users from the micro and meso layer use fixed infrastructures such as base station or other internet access points to communicate with the macro cloud layer. Macro layer consists of hybrid cloud, where important traffic data can be stored in a secured private cloud and system upgrade or operations can be carried in the public cloud. Hybrid cloud structure accelerates the deployment of traditional IT resources in the cloud model. They also present a conceptual model to enrich mobile GIS functions with the integration

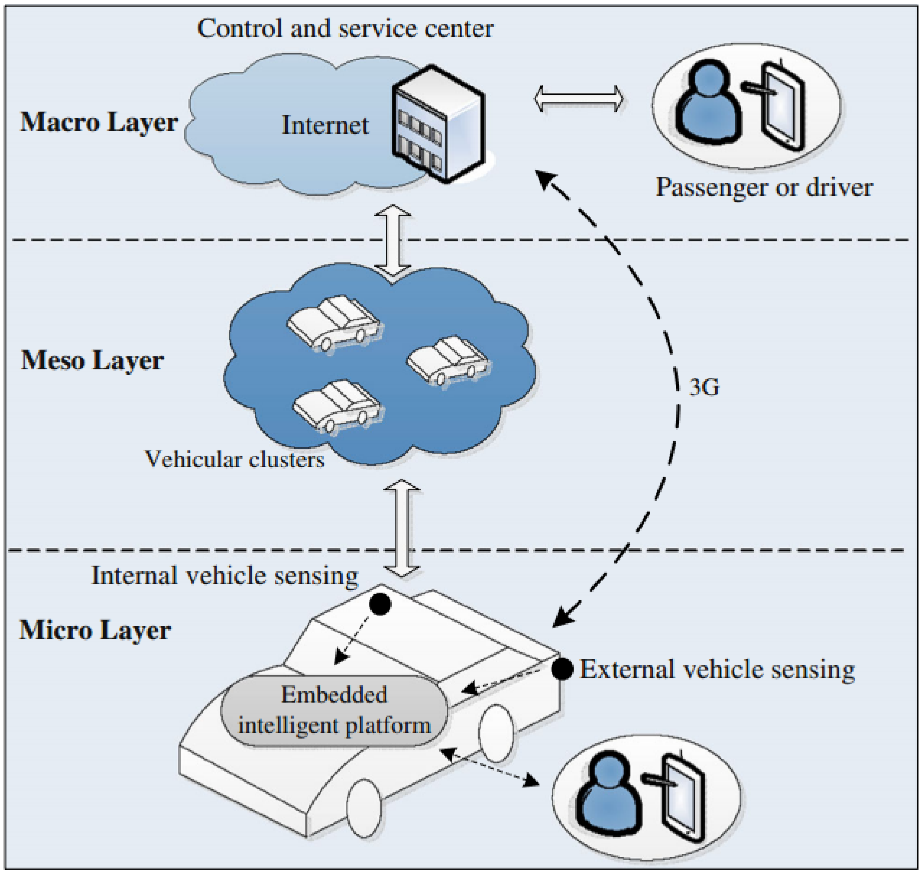


Figure 2.10: Hierarchical vehicular CPS described in [33]

of traffic cloud data and suggest to use ArcGIS[1] and UNETRANS[2] model. In addition, they present a conceptual traffic prediction model to facilitate dynamic vehicle routing based upon historical traffic data, real-time traffic data and traditional traffic department data.

2.2.3 Analytical models of VCPS

In [59], the authors develop a finite queue model to analyze queue occupancy distribution at junctions and traffic signals. They measure waiting time distribution and traffic congestions to find the shortest path and to receive early jam alert. In [60], the authors propose an approach to avoid broadcasting storm in VANETs by prioritizing a subset of locally

[1]https://www.arcgis.com/
[2]http://www.ncgia.ucsb.edu/ncrst/research/unetrans/first.html

generated messages over distant messages for the safety applications. In their model, they attempt to minimize the critical distance between the event location and the vehicle that would receive and react to the event information. In [61], authors present intelligent parking service models following birth-death stochastic process, and a vehicular data mining model based on Naïve Bayes model. In [33] authors propose a probabilistic method to analyze vehicle maintenance service response time using cloud based VCPS data so that users can receive timely warning about abnormal vehicle problems.

In [62], the authors modelled connectivity in the vehicular ad-hoc network in highway scenario. They modelled traffic states in terms of vehicle speed, traffic density and traffic flow. The authors first modelled the connectivity using platoon size and connectivity distance metrics and later extended the basic model for complex scenarios such as two direction traffic, multilane highways, and heterogeneous vehicle networks. Their analysis shows that the increase of traffic flow and/or increase of the transmission range is good for connectivity. Whereas, higher vehicular speed reduces the connectivity. Since, IoT is increasingly looking into Sensing-as-a-Service model, we study the mathematical model of [63] that measures the performance of web service composition. Their proposed model can predict web service utilization changes, the duration to complete new process calculations, as well as optimize the *Service Level Agreements* of the service providers and the service integrators.

Again, from the data perspective, safety message dissemination about a hazardous condition was studied in [64], where the authors measure the performance in terms of average delay to propagate the message, the number of nodes receiving the relay information and the number of duplicate messages received by each vehicle. Data dissemination algorithms with time-probabilistic characteristics are described in [65], where the authors provide a simplified model of the dissemination delay. In [66], the authors provide analytical study on the performance of data dissemination in VANETs. They divide the traffic in two priority classes, higher priority and lower priority. The performance is measured in terms of average message forwarding distance. Their analysis proves that the increase of transmission range improves message forwarding, but the improvement is limited due to internal and external interferences at the receiver node.

2.3 Discussion

The evolution of *Ubicomp* has seen the wave of research trends moving from WSN to the IoT [42], which eventually created a new paradigm of thoughts SIoT [38, 67]. As one of the key elements of the *Ubicomp*, vehicles have also seen a lot of research interests for effective deployment of the ITS. VANETs has received some dedicated focus for a while to build advanced communication infrastructures for vehicles ([20, 21, 22, 23, 24]), which recently shifting towards the Internet of Vehicles (IoV) [25, 26, 27].

CPS is considered as the source of new industrial revolution [68], where IoT will be the foundation to realize the CPS. This revolution will have a great impact on the society and overall world economy. In this new phenomenon, machines will easily interact with other machines and humans. The combination of artificial intelligence (AI) with cloud infrastructure, IoT and machine learning will ensure that machines understand the needs of humans and act accordingly. As CPS includes traditional embedded systems and controls, the adoption of IoT as CPS foundation will introduce new approaches to the CPS implementations [69]. Considering on going development of the CPS concepts, Vehicular CPS (VCPS) provides ample scopes for further developments. Contemporary researchers are taking their motivations either from VANETs, IoT, or cloud computing to formulate VCPS architectures. In this dissertation, we accumulate our contributions focused on the field of VCPS.

In Chapter 2 we present a comprehensive architecture for cloud based CPS (C2PS), where we use the standard CPS design concepts to incorporate cloud supports to it. In Table 2.1 we compare relevant works that present CPS architecture models. From the table, we see that researchers mostly described the integration of CPS and cloud supports (i.e. C2PS) using descriptive models, which lack formal description of the three key characteristics of a CPS: computation, communication and control. In our work, we have followed the state machine based analytical design techniques to describe this integration. In this process, we have identified various types of computations and communications (i.e. physical, cyber and hybrid) possible in the C2PS. We also present bayes network and fuzzy logic based reconfigurable model that considers system contexts while selecting a possible interaction mode. This kind of smart connection model has been prescribed for the CPS in [32]. Additionally, we also present a model to describe the formation of various possible cloud infrastructures.

We proposed a VCPS architecture (Chapter 4) [70][71][72] that uses individual vehicle's computing resources to determine vehicular events using internal and external vehicular or smartphone sensors. This information is shared among vehicular platoons using either DSRC or 3G/LTE based communication methods. Our architecture supports both direct V2V or SenAS based cloud assisted P2P data communications, which enables both real time safety and delay tolerant non-safety applications. In order to leverage IoT, we derive our motivations from SIoT and proposed a VCPS architecture named *SIoV*. *SIoV* identifies M2M social relations and related interactions among VCPS subsystems; reuses VANETs components and maps them to the IoT Architecture reference model; organizes VCPS interactions using a message structure, *tNote*; and supports different levels of data clouds.

We study the effects of data workloads on individual systems that are involved in a VCPS and how traffic parameters affect them [73]. The models developed in this study are useful to monitor the workloads dynamically at different ITS scenarios. We validate the proposed workload models using extensive simulations and find the relationships among the system parameters that are used to design reconfiguration approaches for VCPS subsystems. This would help to measure data loads on various levels of clouds and would foster real life VCPS deployments (Chapter 5).

Details about C2PS application development platform are presented in Chapter 6. The server side of the platform has been developed using JAVA, MySQL, and JSON technology. Whereas the client side of the application platform is developed using Android technology and can be deployed to mobile devices such as Smartphones and Tablets. We also build a VCPS emulator and other VCPS applications using the developed platform [74].

Related Works	Cloud Integration	Computation	Communication	Control	Configuration	Model Type
Future Internet of Things [35]	Centralized data center	Inter and Intra CPS	Authority wise IoT infrastructure network	Distributed control node	Man like neural network	Descriptive
SIoT [37][38]	Inherently present	Inter and Intra CPS	Service discovery based	Owner based object relationship control	Service composition	Descriptive
Automotive CPS [50]	N/A	Vehicle Intra CPS	Inter-linked embedded systems	Power distribution control	Master-Slave based	State Machine Based
Multi-tier CPS [51]	Software-as-a-Service	Multi-tier based	Inter tier communication	Cloud based control algorithms	N/A	Descriptive
VCMIA [33]	Inherently present	Layered computing	Usual VANETs communication	N/A	QoS and Data mining	Descriptive
VANET-Cloud [34]	Conventional Cloud (IaaS, PaaS, SaaS) and Temporary Cloud	Layered computing in VANETs	Permanent and temporary cloud, Client to data center using various means	N/A	N/A	Descriptive
V-Cloud [30]	IaaS, PaaS, SaaS	Inter, Intra CPS for vehicle	Vehicle to Vehicle/Infrastructure	Cyber-based context awareness	N/A	Descriptive
VSSA [55]	Software-as-a-Service	Layer based computation	Service interaction	Context based	Service composition	Descriptive
5C Architecture [32]	Cognition functionality	Inter, Intra CPS	Smart connection	Cyber-based	Cyber-based	Descriptive Model
Proposed Cloud based CPS Reference Model	**Inherently present**	**Inter, Intra CPS, Smart**	**Social network like relationships among things and further complex thing formation**	**Bayes network based context awareness**	**Fuzzy rule based**	**State machine and Fuzzy rule based**

Table 2.1: Comparison of state-of-the-art works and proposed cloud based CPS reference model.

Chapter 3

Cloud-based CPS Architecture Reference Model

Research community is showing tremendous interests about the CPS field these days. The contemporary research in CPS is mostly focused in the physical layer of embedded systems or application possibilities of the CPS domain. It is important to provide a clear bridge, how the embedded systems of the physical layers will be leveraged to provide both real time and delay tolerant services to the application layer of CPS. Since cloud infrastructure is becoming abundant in our day-to-day life, we propose to incorporate cloud infrastructure as the intermediate bridge between the physical layer and the application layer. As a result, CPS application design, reconfiguration and smartness become inherently scalable.

In this chapter, we describe an architecture reference model for the cloud-based cyber-physical system (C2PS), where every physical *thing* accompanies a hosted digital twin, a cyber *thing* in the cloud. Two things can communicate either through direct physical connections or through the indirect cloud based virtual peer-to-peer (P2P) connections. We present the C2PS architecture details in Section 3.1, detailed analytical models of the C2PS in Section 3.2, and describe the vehicular CPS using the reference model in Section 3.3.

3.1 Proposed C2PS Architecture

In the proposed C2PS, we assume that a number of independent systems connect together to perform a common goal, where network connections are omnipresent. In C2PS every physical *thing* is automatically accompanied by a representative cyber *thing* hosted in the cloud. There exists a direct one-to-one connection between every twin cyber-physical *thing* (Figure 3.1). Whenever the physical world changes for a physical sensor, it attempts to update the current status to its digital twin (i.e. cyber thing) hosted in the cloud. Every physical *thing* or its corresponding cyber *thing* manages a *Data Store*. Every physical or cyber thing is identified by a unique ID (i.e. IPv6, Universal Product Code (UPC), Electronic Product Code (EPC), etc.) and is aware of the existence of its twin counter part.

Sensors owner has the authority to control the privacy policy of a sensor by granting access to the sensors through the services' middleware layer. Based on the networking or communication criteria set by the owner of the *things*, either the physical or the cyber *things* can create communication groups. Every communication group is identified by a *Relation ID*. All the communications in a particular relationship are transferred to the members of that group only. Any *thing* can be a member of multiple relationship groups at any given time. These communication groups are created as peer-to-peer networking groups in the *Peer-to-peer Relation Layer* of the cloud hosted cyber objects.

The sensory information collected by the physical layer is stored in its own data store and also in the data store of the cloud based cyber layer. Interactions among the *things* can occur either through direct ad-hoc communication (e.g. Vehicular Ad-hoc Networks (VANET) in the physical layer or through the cloud layer using peer-to-peer communications among the hosted cyber objects. Important interaction information is stored by both physical and cyber layer. Whenever an interaction is received through the cyber layer, it is updated to the responsible physical sensor if possible. Similarly, interactions received through the physical layer are transmitted to the cyber layer.

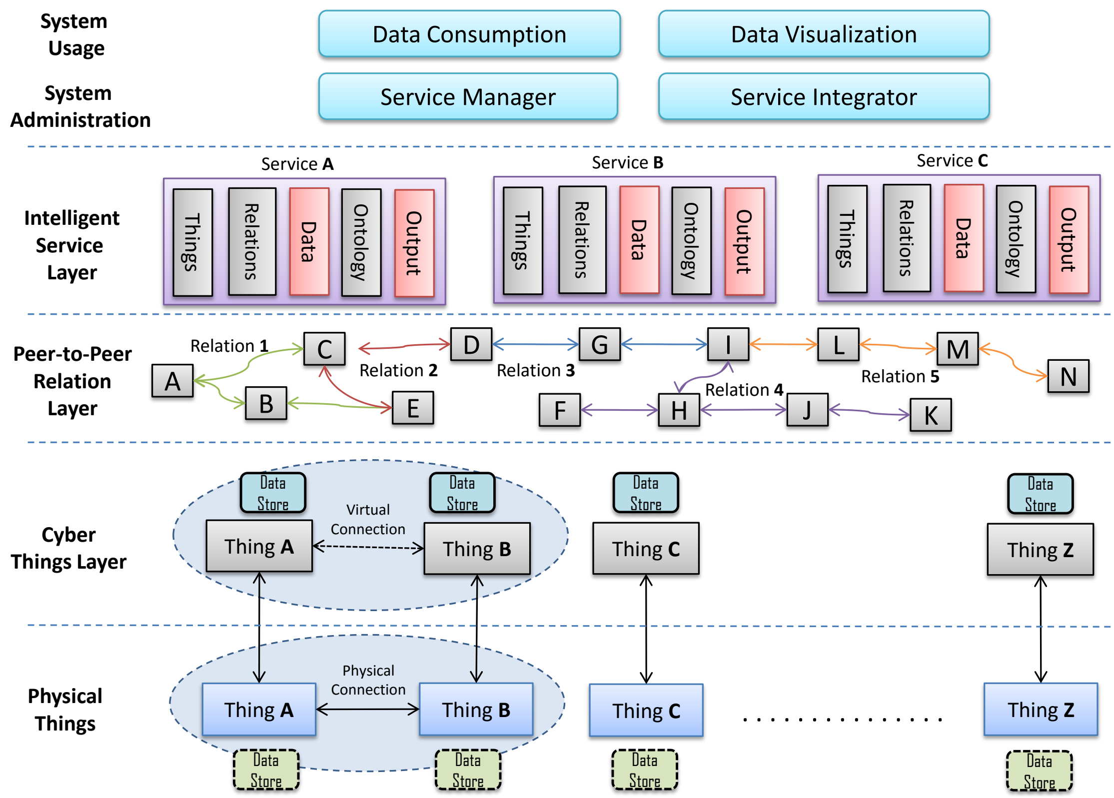


Figure 3.1: Proposed cloud based cyber-physical system architecture.

The proposed architecture adheres to the *SenAS* model, where the data are generated by the *things* and are finally consumed by the humans or by other machines. All the data that are useful to improve the Quality of Service (QoS) of the physical things are stored in the cloud based *Data Center* (Figure 3.2). In the C2PS, a smart thing can be both stationary or mobile and provides some services to other smart things. All the data gathered by the smart things are stored at different levels of storage medium from mobile, stationary to the data center. Interactions in C2PS can be between two mobile things, one mobile and one stationary thing, mobile thing to the data cloud directly, or from one stationary thing to the data center directly. Our proposed cloud model captures all these different combinations of data cloud.

The *Intelligent Service Layer* acts as the middleware, where the cyber *things*, their active relations, and the related ontologies are coupled together. Critical understanding of the low level messages and required control actions by other receiver things can be measured based on domain specific ontologies such as the description onlogy [75], the device ontology [76], etc. Low level reconfiguration of the C2PS is initiated by the service layer. Since data output from the intelligent reasoning stage of one CPS would be consumed across different CPS domains, ontology based data formatting ensures seamless integration of the CPS services.

In C2PS, every smart thing can provide a set of services based on its current capabilities. For example, physical systems in the C2PS can offer real time services, when the communication channel is active through the physical layer, or power supplies of the smart things are sufficient enough to support heavy duty operations, or interacting systems are in their physical communication range. Similarly, a smart thing can dynamically decide to communicate through the cyber layer, when physical capacity is down. Cyber layer objects can offer near real time or delay tolerant services. Since network communication is assumed to be omnipresent, hence a hybrid case of sensor-services fusion is also possible in the C2PS architecture. Data center cloud of the C2PS can provide summary related data mining services. Different combination of services cloud is formed in the C2PS based on the interactions of subsystems (i.e. physical level sensor cloud, cyber level services cloud and hybrid sensor-services fusion cloud).

In the proposed system, data is consumed as reports or as input to other systems. Visibility and data privacy of the *things* are managed through the *Service Manager*. The owner of a *thing* can decide which part of the generated data would be shared publicly, which part would use subscription model or which part will be completely private to the owner. Access to the physical *thing* can be easily cut off from the outside world by switching off the digital twin access rights. Multiple services can also be grouped together to form mashup service for the *Intelligent Service Layer* using the *Service Integrator* component.

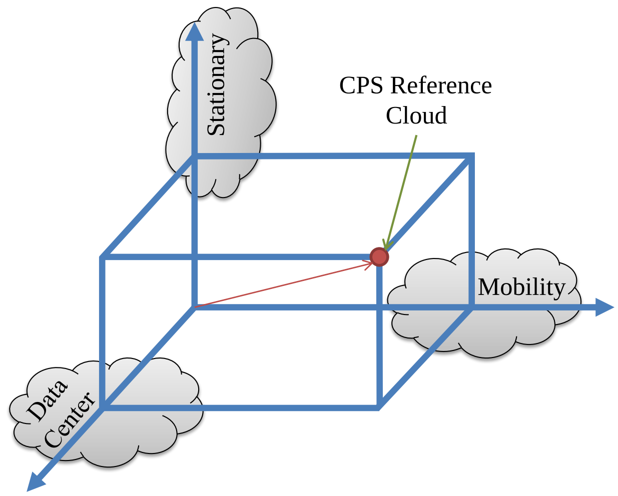


Figure 3.2: Three dimensional cloud structure for the proposed C2PS architecture.

3.2 C2PS Design Details

A CPS is composed of various other independent systems, which can be simple (i.e. composed of a few subsystems) or complex (i.e. composed of many subsystems). The key properties of a true CPS are computation, communication, and control [77]. In the proposed model, we elaborate these key properties for a cloud-based CPS (i.e. C2PS) while we integrate cloud with the CPS. In case of C2PS computation, we derive the types of things/operation modes that are formed in a C2PS. For the control property, we describe how to select one of these operation modes based on the current system context (i.e. smart connection). The communication property describes how complex things (i.e. system of systems) can be formed by communication/interaction of the C2PS subsystems.

From here on we address the independent systems of a C2PS as subsystems and a complex thing as a collection of independent things or subsystems. The proposed C2PS, $\mathbb{S}$, consists of the subsystems: physical things $\mathbb{P}$, a digital twin representative for each physical thing $\mathbb{C}$, hierarchy based composition of subsystems to form further complex things $\mathbb{M}$, relationship network among things $\mathbb{R}$, integration of web services $\mathbb{V}$. Here, $\mathbb{S} = (\mathbb{P}, \mathbb{C}, \mathbb{M}, \mathbb{R}, \mathbb{V})$.

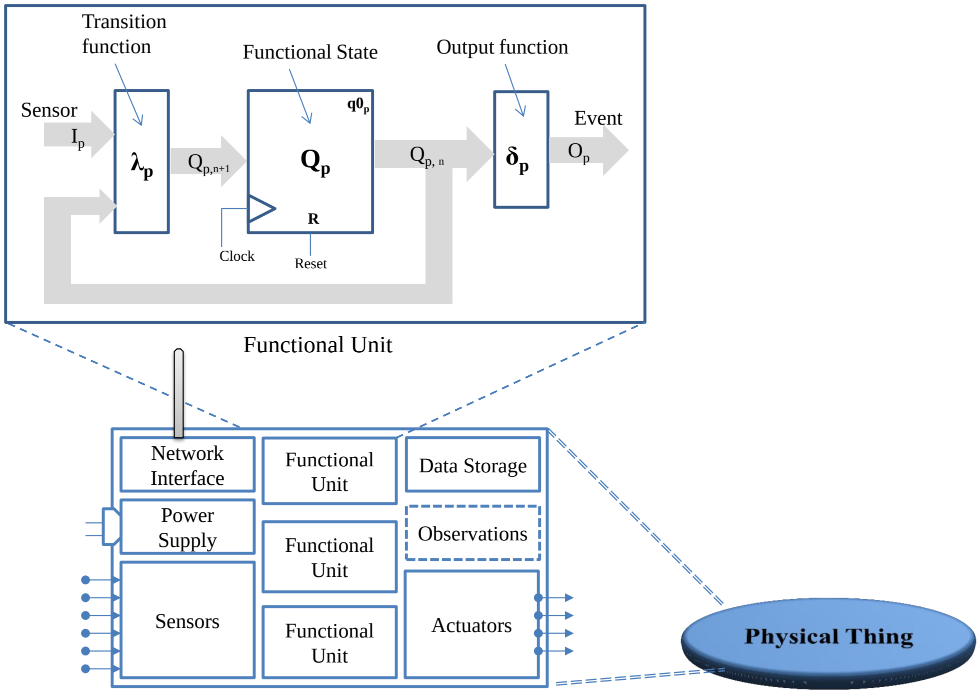


Figure 3.3: Architecture of a physical thing

3.2.1 Computation

In this section we describe the computation property, where we analytically model different operational modes using Moore's [77] finite state machine.

Physical Things ($\mathbb{P}$)

We consider that every physical *thing* $p \in \mathbb{P}$ is comprised of a 7-tuple of elements (S_p, A_p, F_p, E_p, N_p, P_p, D_p) where sensors S_p act as the input to the system and there is at least one functional unit in F_p to process the sensory values to identify events E_p and store the result in the data storage set D_p. Actuators A_p are used to perform action on the environment, and there is at least one network interface in N_p, and one power supply in P_p (Figure 3.3).

$$\mathbb{P} \equiv \{p_i, i = 1...|\mathbb{P}|\} \tag{3.1}$$

Here we adopt the model of [78] to represent every functional unit $f_p \in F_p$ as a sequential finite state machine of 6-tuple,

$$f_p = (Q_p, I_p, O_p, q0_p, \lambda_p, \delta_p) \tag{3.2}$$

Where Q_p represents the states one function is comprised of and $q0$ is the initial state of computation. Various sensor values act as inputs $I_p \subseteq S_p$ to a state that initiates a transfer function λ_p (Equation 3.3) to the other states. Every state has an associate output O_p based on the output function δ_p (Equation 3.4) such that event $O_p \subseteq E_p$ is identified.

$$\lambda_p : Q_p \times I_p \rightarrow Q_p \tag{3.3}$$

$$\delta_p : Q_p \rightarrow O_p \tag{3.4}$$

Cyber Things ($\mathbb{C}$)

In the proposed C2PS system, every physical thing is represented by a cloud based digital twin (Figure 3.4), $c \in \mathbb{C}$, that has seven elements. Here, $c = (S_c, A_c, F_c, E_c, N_c, P_c, D_c)$ consists of virtual sensors S_c, virtual actuators A_c, functional units F_c, event observations E_c, virtual interfaces N_c, virtual power supply P_c and data storage D_c. Here, virtual interface is the communication medium of the cyber thing with the physical thing and virtual power supply indicates that the cloud based process can be easily installed or removed from the cloud.

Virtual sensors of a cyber thing are the output observations of the physical thing. These observations can either be event or direct raw data which will be processed by the functional unit of the cyber thing to identify the observations in the cloud. Equation 3.6 represents the relationship of a cyber thing to a physical thing. A physical thing can always

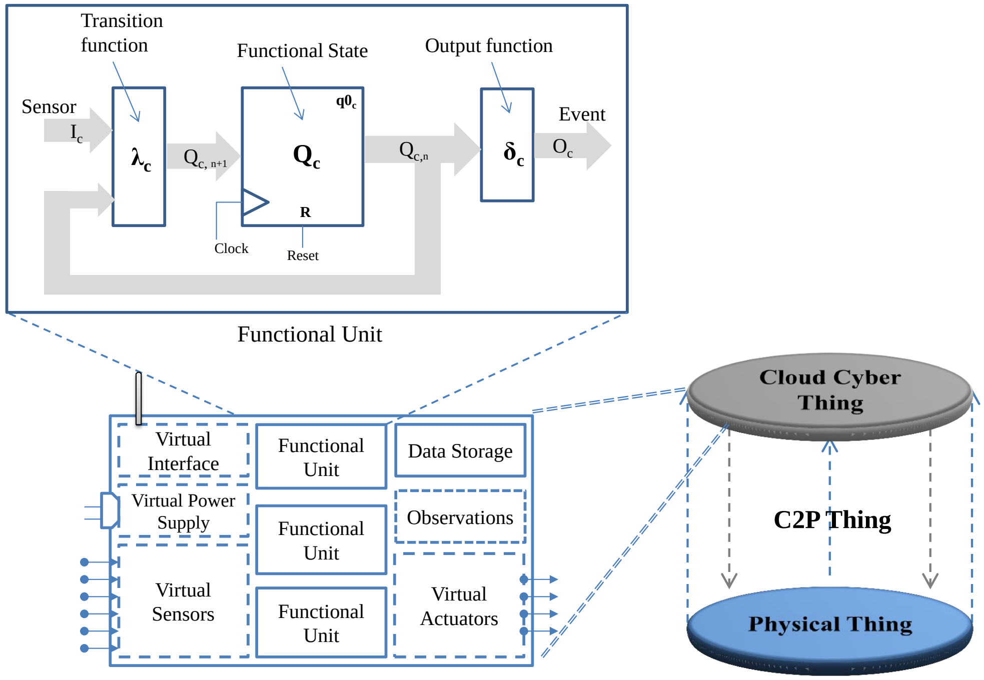


Figure 3.4: Architecture of a cloud-based digital twin (i.e. cyber thing)

perform operations without the help of a cyber thing. Whereas, the opposite $\Delta_c^{-1} \neq \Delta_c$ i.e. replacing a physical thing completely by a cyber thing is not possible. A cyber thing increases the capacity of a physical thing. So, there should be at least one cyber thing for each physical thing, $|\mathbb{C}| \geq |\mathbb{P}|$ in a C2PS. Sequential finite state machine Equations 3.7, 3.8, 3.9 are equally applicable for the cyber things [78].

$$\mathbb{C} \equiv \{c_j, j = 1...|\mathbb{C}|\} \tag{3.5}$$

$$\Delta_c : \mathbb{P} \rightarrow \mathbb{C} \tag{3.6}$$

$$f_c = (Q_c, I_c, O_c, q0_c, \lambda_c, \delta_c) \tag{3.7}$$

$$\lambda_c : Q_c \times I_c \rightarrow Q_c \tag{3.8}$$

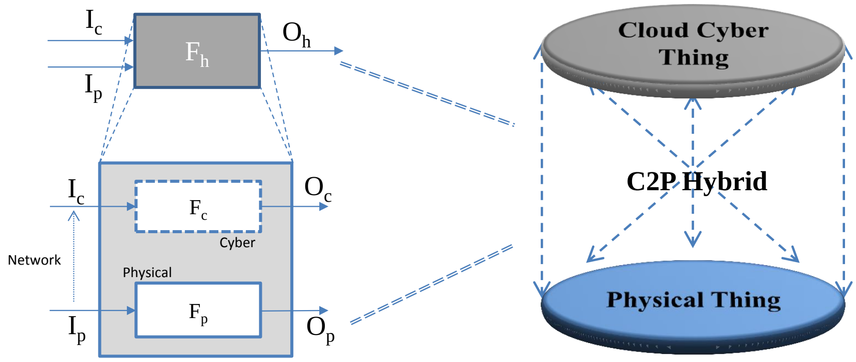


Figure 3.5: Architecture of a cloud-based hybrid cyber-physical thing

$$\delta_c : Q_c \rightarrow O_c \tag{3.9}$$

Advantage of installing a cyber thing for each physical thing is that even a low profile device, which now only acts as a mere sensor source can become smarter without much physical changes by only developing the cyber counter part. And the cyber things can also be organized hierarchically to develop further smarter things that can share important information about a physical region.

Hybrid Things ($\mathcal{H}$)

From the above description of the physical and the cyber things (Equation 3.2, 3.7), we can formulate a hybrid cyber-physical thing, $h \in \mathcal{H} \equiv (S_h,\ A_h,\ F_h,\ E_h,\ N_h,\ P_h,\ D_h)$, where part of the computations (i.e. low cost) occur in the physical layer and the rest of the computations (i.e. higher cost) occur in the cyber layer. It assumes that the network communication cost is negligible and some physical sensors act as inputs to the cyber things while computing independently. This subsystem can be described as synchronous side-by-side composition of machines, where $Q_h = Q_p \times Q_c$ are states, $I_h = I_p \times I_c$ are inputs, $O_h = O_p \times O_c$ are outputs, $q0_h = (q0_p, q0_c)$ is the initial state and λ_h, δ_h are transfer (Equation 3.11) and output (Equation 3.12) functions respectively [77][79].

$$f_h = (Q_h, I_h, O_h, q0_h, \lambda_h, \delta_h) \tag{3.10}$$

$$\begin{gathered}\lambda_h : (Q_p \times Q_c) \times (I_p \times I_c) \rightarrow (Q_p \times Q_c) \text{ where,} \\ \lambda_h((q_p, q_c), (i_p, i_c)) = (\lambda_p(q_p, i_p), \lambda_c(q_c, i_c))\end{gathered} \tag{3.11}$$

$$\begin{gathered}\delta_h : (Q_p \times Q_c) \times (I_p \times I_c) \rightarrow (Q_p \times Q_c) \text{ where,} \\ \delta_h((q_p, q_c), (i_p, i_c)) = (\delta_p(q_p, i_p), \delta_c(q_c, i_c))\end{gathered} \tag{3.12}$$

3.2.2 Control

In case of the C2PS, we consider every sensing or actuation request as a physical event. For any physical sensing event, a thing is involved in data collection, computation and transmission. Here transmission means data sharing with other connected things that are in a relationship (r) with the data producer. There can be three modes of computations and subsequent interactions possible in a C2P thing (Figure 3.6): physical-physical, cyber-cyber, and cyber-physical hybrid. An important control application is to select one of these computation-interaction modes that can be regarded as context aware self-reconfiguration.

Context Aware Self Reconfiguration

In the `physical-physical` interaction of a C2PS, all the computations occur in the physical thing and the data sharing occurs through the direct physical communication channel (Figure 3.6.a). For the `cyber-cyber` type of interactions, all the computations occur in the cloud layer and the interactions among the things occur through the cloud-based cyber layer. All computation updates are notified to their respective physical things from their corresponding cyber things (Figure 3.6.b). The other type of interaction is `cyber-physical`, where the computations are splitted in both the physical and the cyber layers and data sharing also happens simultaneously in both the layers. At the end of

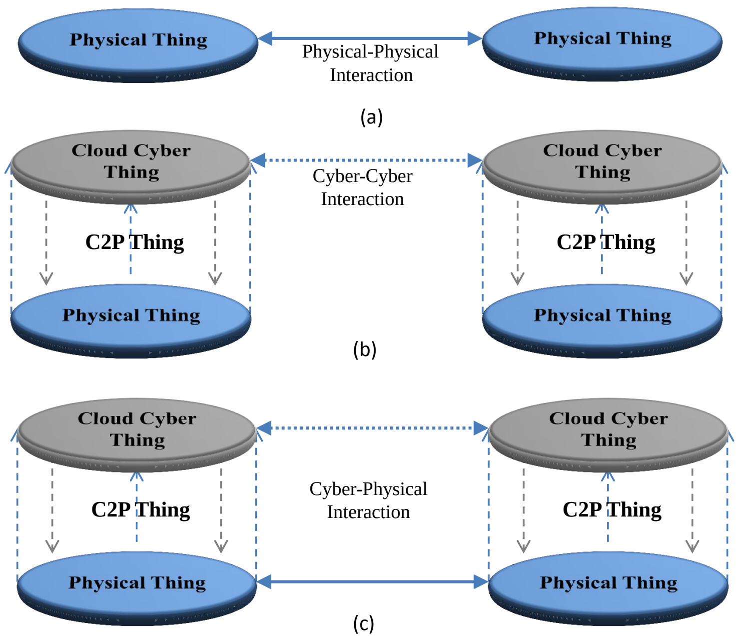


Figure 3.6: Three types of computations and interactions possible between two C2P things: a) physical-physical interaction, b) cyber-cyber interaction, and c) mixed cyber-physical interaction

each operation session, physical layer of a thing is updated with the results of the cyber layer and vice versa (Figure 3.6.c). A smart C2PS thing can automatically decide to select any of these modes considering current system contexts. The probabilistic framework of Bayesian networks (BNs) is a popular choice to model uncertainty of context awareness for a long time [80][81][82], which motivates us to select BNs to design the smart connection controller.

Bayesian Network-based Context Model

At any moment, a smart thing can choose either of the three operation modes based on the current system contexts. Here we assume, the contexts of a smart *thing* at any time can be battery power, computation cost, communication cost, communication range, etc. The decision system uses a Bayesian network represented as a directed acyclic graph (DAG) in Figure 3.7.

In the DAG model, the immediate contexts of a thing are considered for example as

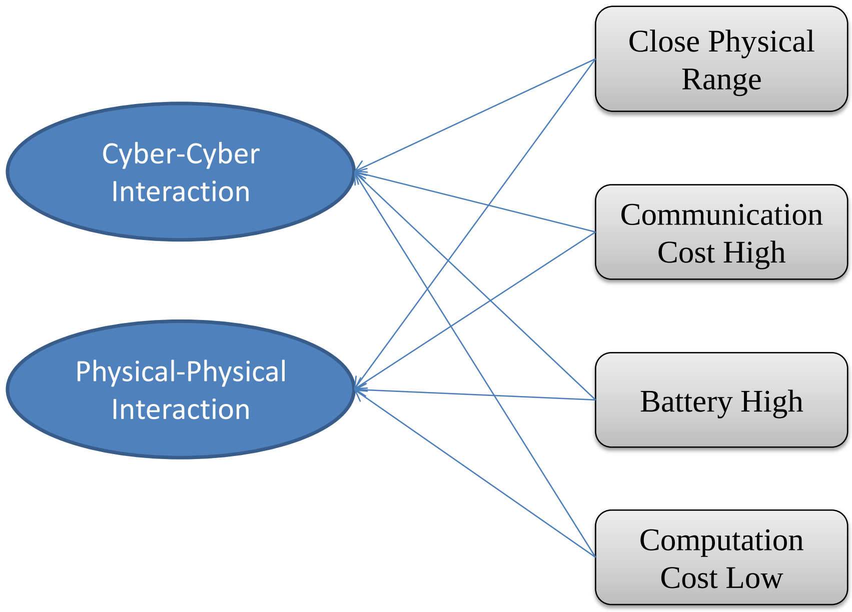


Figure 3.7: DAG model of context aware interaction controller showing causal influences.

"*close physical range*", "*communication cost high*", "*computation cost low*" and "*battery high*" that are represented as Cr, Ch, Cl, and Bh respectively. The direct complement of these events can be described as "*out off range*", "*communication cost low*", "*computation cost high*" and "*battery low*" which are represented as Cr', Ch', Cl', and Bh'. Here, $Cr' = (1 - Cr), Ch' = (1 - Ch), Cl' = (1 - Cl),$ and $Bh' = (1 - Bh)$. In order to select the `physical - physical` communication mode, we consider the contexts Cr, Ch, Cl, and Bh. Similarly, a thing can choose the `cyber - cyber` mode based on the contexts Cr', Ch', Cl', and Bh'. Since the proposed DAG (Figure 3.7) is a polytree (i.e. singly connected network) [83], hence the joint probability distribution of any node can be represented as [84],

$$\begin{aligned} P(X_1 = x_1, ..., X_n = x_n) &= \prod_{i=1}^{n} P(X_i = x_i | X_{i+1} = x_{i+1}, ..., X_n = x_n) \\ &= \prod_{i=1}^{n} P(X_i = x_i | X_j = x_j \text{ for each } X_j \text{ which is parent of } X_i) \end{aligned} \tag{3.13}$$

Here, the conditional probability of selecting `physical - physical` communication mode is,

$$P(PP|Cr,Ch,Cl,Bh) = \frac{P(PP,Cr,Ch,Cl,Bh)}{P(Cr,Ch,Cl,Bh)} \tag{3.14}$$

Also, the conditional probability of selecting `cyber - cyber` communication mode is,

$$P(CC|Cr,Ch,Cl,Bh) = \frac{P(CC,Cr,Ch,Cl,Bh)}{P(Cr,Ch,Cl,Bh)} \tag{3.15}$$

Once we have the joint probability distribution values from Equation 3.13, we can find the probabilities of selecting `physical - physical` and `cyber - cyber` modes. We further use them as inputs to a fuzzy logic decision system that can select either of them or the `cyber - physical` option. As `cyber - cyber` is the opposite mode of the `physical - physical` communication mode, we use a fuzzy logic based decision system to describe the intermediate ranges. Also, fuzzy logic rules can be easily updated to instill higher degree of reconfiguration in the control method.

Fuzzy Logic based Controller Model

Figure 3.8 shows the architecture of a fuzzy logic based control system, where the input I_c is first fuzzified to I_c^f and after the rule base association and inference application, it generates output O_c^f which is later defuzzified to O_c [85]. We have selected fuzzy logic since its rule base can be always updated, which suits the nature of C2PS hybrid computing reconfiguration.

$$I_c^f = \{(i_c, \mu^f(i_c))|i_c \in I_c\} \tag{3.16}$$

Here every fuzzy input is an ordered pair of the input (i_c) and its grade of member function ($\mu^f(i_c)$). Member functions can be of type triangular, gaussian, bell-shaped,

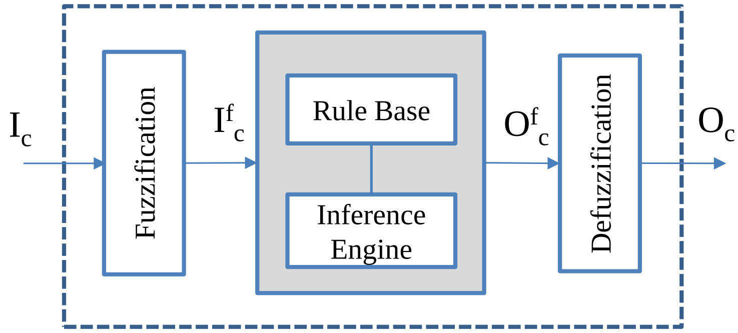


Figure 3.8: Abstraction of a basic fuzzy logic based decision system.

sigmoid, polynomial, etc [86]. The rule base of a Multiple Input Single Output (MISO) fuzzy system can be written as,

R_1: if i_1 is A_1 and i_2 is B_1 then o_1 is C_1

R_2: if i_1 is A_1 and i_2 is B_2 then o_1 is C_2

R_3: if i_1 is A_2 and i_2 is B_1 then o_1 is C_3

R_4: if i_1 is A_2 and i_2 is B_2 then o_1 is C_4

Here i_1, i_2 are sensor variables and o_1 is an output variable respectively. A_i, B_i, and C_i are linguistic values of the linguistic variables i_1, i_2, and o_1 in the universe of discourse of W, X, and Y respectively. Here, a fuzzy control rule such as R_1 can be defined as,

$$\begin{aligned} \mu_{R_i} &\equiv \mu_{(A_i \text{ and } B_i \rightarrow C_i)}(w, x, y) \\ &= [\mu_{A_i}(w) \text{ and } \mu_{B_i}(x)] \rightarrow \mu_{C_i}(y) \end{aligned} \tag{3.17}$$

Where A_i and B_i is a fuzzy set $A_i \times B_i$ in $W \times X$; $R_i \equiv (A_i \text{ and } B_i) \rightarrow C_i$ is a fuzzy implication (relation) in $W \times X \times Y$ space. Each of the fuzzy relation represents a fuzzy logic controller. The values of W, X, and Y are selected based on the new smart *thing* to be designed. The output of the fuzzy relations can be defuzzified using centroid of area, mean of maximum, bisector of area, etc. [87]. In the following section we design a MISO controller to select one of the interaction modes.

Interaction Controller Design

In order to select one of the communication modes, we design a MISO controller, where there are two inputs $P(PP|Cr, Ch, Cl, Bh)$, $P(CC|Cr, Ch, Cl, Bh)$ to the fuzzy interaction model and one output is selected from $\{\rho_{cc}, \rho_{pp}, \rho_{cp1}, \rho_{cp2}\}$ (Figure 3.9). The rule base matrix can be seen in Figure 3.10. As an example, we divide the probability space as *Very Low (VL), Low (L), Medium (M), High (H),* and *Very High (VH).* Here the input space for `physical - physical` and `cyber - cyber` is $W = \{VL, L, M, H, VH\}$ and $X = \{VL, L, M, H, VH\}$ respectively. The output space is again considered as *physical-physical (PP), cyber-cyber (CC), cyber-physical-1 (CP1),* and *cyber-physical-2 (CP2),* where $Y = \{\rho_{pp}, \rho_{cc}, \rho_{cp1}, \rho_{cp2}\}$. The input and output space can be configured according to the needs of a system manufacturer.

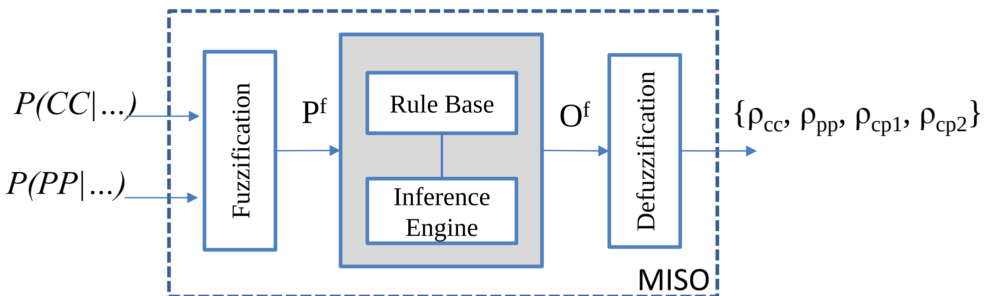


Figure 3.9: Fuzzy logic based smart interaction type selection model. It is a MISO (Multiple Input Single Output)

3.2.3 Communication

Multiple C2PS things can work as subsystems of a further advanced C2PS ($\mathbb{M}$). The interaction mode of a C2PS subsystem is transparent to other C2PS subsystems. We can build an advanced system through the cyber layer, where a cyber thing (i.e. digital twin) communicates with other cyber things by following a topology or relationship. For simplicity, we take the total number of possible advanced systems to be the power set of $\mathbb{C}$, $\mathcal{P}(\mathbb{C}) = 2^{|\mathbb{C}|}$. Every advanced thing is denoted by an unique Id $\mathcal{T}$ so that $\mathbb{M}_{\mathcal{T}}^{-1}(c)$

Physical-Physical

Cyber-Cyber	VL	L	M	H	VH
VL	CP2	CP2	PP	PP	PP
L	CP2	CP1	CP1	PP	X
M	CC	CP1	CP1	X	X
H	CC	CC	X	X	X
VH	CC	X	X	X	X

Very Low (VL) = 0.000 - 0.250
Low (L) = 0.175 - 0.425
Medium (M) = 0.350 - 0.650
High (H) = 0.575 - 0.825
Very High (VH) = 0.750 - 1.000

Physical-Physical (PP) = 0.000 - 0.500
Cyber-Physical-1 (CP1) = 0.300 - 0.500
Cyber-Physical-2 (CP2) = 0.500 - 0.700
Cyber-Cyber (CC) = 0.500 - 1.000

Figure 3.10: Fuzzy logic rules matrix to select the communication mode. Here each color represents a mode to be selected. Don't care combinations (X) represent the situations that are no possible (i.e. $P(CC) + P(PP) > 1$)

for $c \in \mathcal{P}(\mathbb{C})$ returns an unique $\mathcal{T}$. Each of this master things works as a hub of other networked cyber things. Every network is uniquely tagged by a relationship Id $\mathcal{R}$ and fulfils a specific goal $\mathcal{G}$. The subsequent advanced things fulfill the Equation 3.18;

$$f_r : \mathbb{M}_{\mathcal{T}} \rightarrow \mathcal{R}, \exists g \in \mathcal{G} \tag{3.18}$$

Hierarchical Composition of C2PS Things

A possible structure of complex systems is to organize them hierarchically. In this case, a higher level cyber thing is composed of further lower level cyber things. Here, one higher level thing works as the master of the lower level slave things. And, any state transition to a master thing means a state transition for any of the slave things. We can define such a complex system using hierarchical composition of finite state machines (Equation 3.19), where $\mathcal{T}$ is a set of unique IDs for the cyber things. We follow the methods of [78].

$$\mathbb{M}_{\mathcal{T}} : \mathbb{C} \rightarrow \{\{\emptyset\} \cup \mathbb{C}\},\ \mathbb{M}_{\mathcal{T}}^{-1} : \mathbb{C} \rightarrow \mathcal{P}(\mathbb{C}),\ f_r \tag{3.19}$$

Here we see that every cyber thing can be a composition of other cyber things $c \in \mathcal{P}(\mathbb{C})$. The maximum number of subsets of the $\mathbb{C}$ is $\mathcal{P}(\mathbb{C})$ denotes the power set of the cyber things

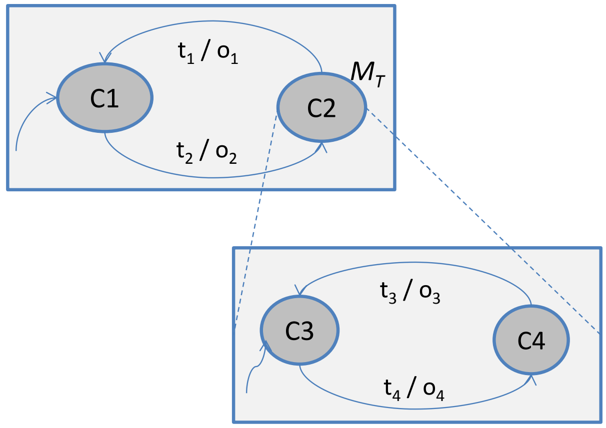


Figure 3.11: Abstraction of hierarchically organized things

that is $2^{|\mathbb{C}|}$. As a result, we might need to create more logical entities of some cyber things to build a complex thing which is easy for C2PS. But in any case $\mathbb{M}_{\mathcal{T}}^{-1}(c)$ will return the unique $\mathcal{T}$ and related $\mathcal{R}$.

If we represent Equation 3.8 as Λ_c and Equation 3.9 as Δ_c then we can assume mean $\bar{\mathbb{C}}$ and define transfer and output function as Equation 3.20 and 3.21 respectively [78],

$$\Lambda_{\mathbb{M}_{\mathcal{T}}^{-1}(\bar{\mathbb{C}})} \equiv \prod_{i=1}^{\mathbb{M}_{\mathcal{T}}^{-1}(\bar{\mathbb{C}})} \Lambda_i \tag{3.20}$$

$$\Delta_{\mathbb{M}_{\mathcal{T}}^{-1}(\bar{\mathbb{C}})} \equiv \prod_{i=1}^{\mathbb{M}_{\mathcal{T}}^{-1}(\bar{\mathbb{C}})} \Delta_i \tag{3.21}$$

Star Networked C2PS Things

Complex thing can be organized in a star networked topology (Figure 3.12), where a master cyber thing acts as a hub of other cyber things and the outputs of the lower level subsystems are inputs to the higher level system. We can define the system by Equation 3.22, where transfer and output functions are similar to the general synchronous finite state machine.

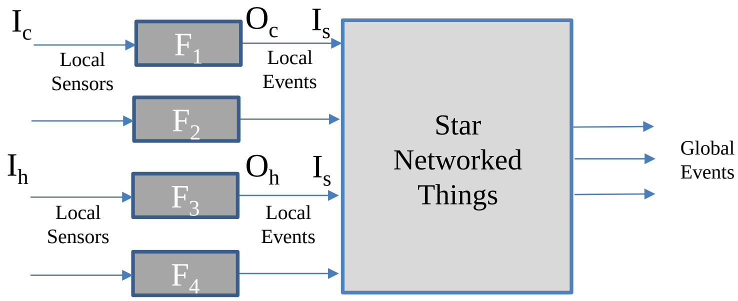


Figure 3.12: Architecture of the advanced star networked things

$$
\begin{aligned}
f_s = (Q_s, I_s, O_s, q0_s, \lambda_s, \delta_s), f_r \text{ where } V_c \subseteq V_s \\
\text{assume, } V_c \text{ is data type of } O_c \\
\text{and, } V_s \text{ is data type of } I_s
\end{aligned}
\tag{3.22}
$$

$$
\lambda_s : Q_s \times I_s \to Q_s \tag{3.23}
$$

$$
\delta_s : Q_s \to O_s \tag{3.24}
$$

This type of advanced thing takes the lower level event information and process them to find further regional or global knowledge. The lower level systems generate the type of information that the higher level system can process. Hence, the higher level system can be replicated and new lower level things can be plugged in as long as data type matches. For example, if we have four smart temperature sensors deployed to four corners of a room then the individual temperature data collected from the four cyber things of the respective sensors can be fed into a higher level master cyber system that recognizes temperature data and can produce an aggregated temperate of a room.

3.2.4 Cloud Services

From the above design models, we see that there can be three types of cloud setup: physical sensor cloud, cyber/virtual process cloud and finally sensor-service integration cloud. The sensor cloud is formed by real world level ad-hoc communication among the C2PS things. Every physical thing involved in a C2PS communication has their own data storage, and communication infrastructure. These storage and network facilities can be accessed in the physical level sensor cloud by ad-hoc network members. So in this level, we have Storage-as-a-Service, Network-as-a-Service or Software-as-a-Service (SaaS) supports. Physical sensor cloud setup can provide real time or near real time services to the physical layer members.

In the cyber process cloud level, we get delay tolerant services that cannot be provided through the physical sensor level. These services take the sensor input from the lower physical layer, take heavy duty decisions using scalable cloud infrastructures and provide services to its own physical level things or to other things through the cyber process peer-to-peer cloud layer. Cyber processes in the cloud layer can be updated, upgraded or can add new functionalities that is accessible from the low level physical layer. Cyber process layer can provide Virtual Network-as-a-Service in order to create relation based networks out of the physical communiation range. We assume that the physical things will use cellular networks (i.e. 3G/LTE) to communication with the cyber layer. Other possible services from the cyber cloud layer are Storage-as-a-Service or SaaS.

The data center can also provide various types of cloud services such as Storage-as-a-Service, SaaS, and Data-as-a-Service, so that various data mining applications can be accessed from the physical layer or by different monitoring authorities.

3.3 Example Scenario: Vehicular CPS

We follow the design philosophy of the above mentioned C2PS architecture reference model to describe vehicular CPS system, Social Internet of Vehicles (SIoV) (Figure 3.13). SIoV is a type of C2PS, where physical systems involved in VANETs communicate with their twin cyber entities, similar to the virtual sensors described above. This middleware like virtual computing layer maintains one-to-one feedback loop with the real world physical processes. Service request from the cyber layer affects physical sensing or actuation and *vice versa*. This sensor processing and following feedback belongs to intra-CPS feature. The cloud based cyber entities enable virtual peer-to-peer networking among the processes, which fosters heavily decentralized cloud operation (e.g., fog computing [88]). The P2P abstraction of the physical ad-hoc network forms machine-to-machine (M2M) social network among the VANETs systems. As a result, vehicles can communicate with other vehicles either through physical ad-hoc or through virtual layer P2P ad-hoc communications. This feature belongs to inter-CPS communication.

SIoV offer services related to safety, efficiency, comfort and entertainment applications to the VCPS users. The main physical components (*i.e.*, SIoV subsystems) involved in this process are vehicles (V), roadside infrastructures (I), houses (H) and the cloud (C). These physical components also have their representative virtual entities on the cyber layer. Real time safety services can be offered either through the VANETs or the cellular network in SIoV. Non-safety delay tolerant applications use the virtual computing cyber layer. Also, data mining and knowledge driven recommendation applications can use the social graph data cloud.

In SIoV, every physical or virtual component has its unique ID and works as a hub for the sensors and services. The presence of cyber or physical entity in communication is ubiquitous to the consumers. The system takes smart decisions between physical (*i.e.*, *ad-hoc*) or cyber (*i.e.*, 3G/LTE) communication based on the current context. Every vehicle does internal/external sensor based fusion in the physical level to identify events. If it is better to use the virtual computing cyber layer for services purpose, vehicle controller takes that decision based on its current context. Other than the services fusion it is also possible to use sensor-services fusion to extend the capacity of different types of services

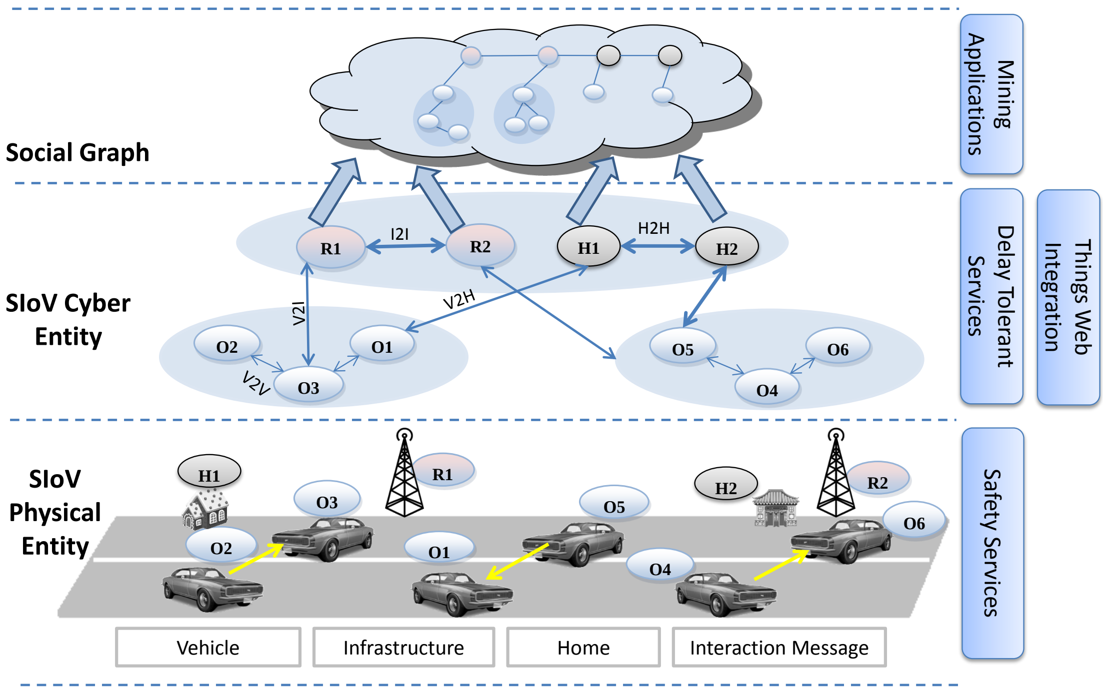


Figure 3.13: Architecture of Social Internet of Vehicles.

of VCPS.

SIoV assumes that a physical vehicle is capable of establishing ad-hoc wireless (e.g., WAVE) connections with other neighbour vehicles while being connected to the central cloud using 3G/LTE type communication technology [25]. In this model, the sensory data or the services are first provided by the ad-hoc wireless channel. If ad-hoc network service is not available, then services are provided from the virtual entities through the cyber cloud. As a result, the services are omnipresent while network communication is ubiquitous. Some of the possible types of interactions among the SIoV subsystems (i.e. vehicle, infrastructure, house, cloud) are V2V, V2I, I2I, I2C, V2H, H2H, and H2C.

The SIoV has two types of data sharing scenarios [70, 72], (1) *Static*: a vehicle is contacting with its home or with its physical friends to share data using V2H or H2H, (2) *Dynamic*: a vehicle is moving on the roadway and sharing information with other moving vehicles and the road side infrastructures using V2V or V2I. Dynamic relationships are temporary and tracked using the time stamp, whereas static relationships are more lasting.

Static friends are those which maintain lasting relationships with the owner of the vehicle. For example, the neighbour houses where the vehicle owner lives or the mechanic shop to which the owner takes his vehicle for servicing or the manufacturer of the vehicle are static friends. Data shared among the static friends are more private than the dynamic data exchanges that occur in the public infrastructure. Private cloud setup can be used for private data exchange. All these relationships are fundamentals for cyber layer P2P network.

It has been recommended in VANETs literature for the vehicles to emit "*i am here*" type messages 10 times/s [89] for safety applications. Other types of interaction messages are infrequent, and they only occur based on specific requests from other vehicles. Every physical vehicle along with its current sensory information is accessible through the cyber layer of P2P relationships. The services offered by every vehicle are consumed either from the physical vehicle or through the cyber computing layer. Services can be free of cost or can have a price tag agreed upon the consumer and the provider. The interactions among the SIoV subsystems are stored as a social graph in the system storage. In the social graph, nodes are the participating physical subsystems (*i.e.*, V/I/H) and edges are the interactions. Important information about an interaction is stored in the edges with their respective time stamp.

3.4 Summary

In this chapter, we present an architecture reference model to build cloud-based cyber-physical (C2PS) systems. At first, we have presented a three dimensional cubic model for the cloud infrastructures that can offer various combinations of cloud services for real world physical sensory layer, cloud based processes and the center data center. Later, we present analytical design of the C2PS. In this phase, we have described the key characteristics of CPS, computation, communication and control for the proposed C2PS.

We describe the computation and communication properties following Moore's machine based state machine design philosophy. On the other hand, we describe the control property using Fuzzy logic based model so that the rules can be always updated. We present a Bayesian network, and Fuzzy logic based controller design to select a possible computation-interaction mode, which ensures a smart thing has system context awareness. We also present a design of vehicular CPS (VCPS) following the C2PS architecture design philosophy.

Chapter 4

Designing VCPS with IoT Foundation

Vehicular Cyber-Physical Systems (VCPS) is a research field that requires a lot of developments. It is the latest trend that was originated from the WSN, then moved to the VANETs, and recently to the IoV. VCPS captures these topics under one bigger and wider umbrella with additional features of cloud computing and control feedback to the real world. Some of the contemporary research works that fall into VCPS category revolves around the human centric vehicular social network (VSN). In this chapter, we move the attention from human centric VSNs to the machine centric SIoT, where *things* are the central social entities. A related publication can be found at [72], where we describe Social Internet of Vehicles (SIoV). SIoV is a VCPS architecture that presents detailed design of VCPS while keeping IoT as the basic building block.

In the SIoV, we have identified the social structures, relationship types, as well as different interactions that occur between VCPS subsystems, involved in vehicular operations (Section 4.1). We leverage existing systems involved in the VANETs operations and extend them to the VCPS functionalities. For this purpose, we follow the standard IoT-Architecture reference model to map existing VANETs systems to the IoT building blocks (Section 4.1). We also present a communication message structure which inhabits both safety and non-safety messages following automotive ontology, SAE J2735 message set and ATIS schema (Section 4.2).

4.1 Social Internet of Vehicles

In order to describe SIoV (Section 3.3), we use some established acronyms from the VANETs model such as OBU (On Board Unit) and RSU (Road Side Unit) [90]. Additionally, we assume that every vehicle also belongs to a household and there exists a Home Base Unit (HBU) to which all the vehicles and household devices are connected to form the IoT. All these units can also connect to the Internet provider, the Cloud.

OBU represents the storage-computing unit of a vehicle, and RSU represents the storage-computing unit of a road side infrastructure that is always connected with the Internet. OBUs can form wireless ad-hoc networks to consume services (safety and non-safety) from other vehicles and RSUs using WAVE (IEEE 802.11p) [90]. At the same time, OBUs are also connected to the cloud using 3G/LTE cellular networks to synchronize the information with the cyber layer. As a result, SIoV is a cyber-physical system using both the ad-hoc and cellular networks.

While a vehicle is travelling from one roadside infrastructure to the other, it generates sensory messages to be shared with neighbouring vehicles (*i.e.*, platoon members). Important V2V sensory messages are stored in the receiving vehicles in a graph like data structure that is called OBU-OBU social graph. Over time, one vehicle passes through many different platoons and observes many V2V interactions. As a result, the OBU-OBU social graph of a vehicle grows and becomes bulky. When a vehicle that is storing the OBU-OBU social graph gets into the communication range of the infrastructure, it tries to hand over the social graph to the RSU using V2I interaction. If this operation is successful, then the transferred OBU-OBU social graph is appended with OBU-RSU social graph of the infrastructure. An infrastructure can receive many OBU-OBU social graphs over time, which are added as subgraphs to the OBU-RSU social graph. In case of an incomplete V2I handover, the transaction is completed in the following infrastructure which is later synchronized to the originating infrastructure using I2I interaction. If no infrastructure is found, then the transaction is completed in a house (e.g., own or friend) using V2H interaction, which leads to OBU-HBU social graph. All the momentary OBU-HBU or OBU-RSU social graph data available in a house or in an infrastructure of a particular region are transferred to a representative cloud, which leads to a cloud based OBU-RSU-HBU social graph.

4.1.1 SIoV Relations and Interactions

Structures and Relationships

SIoV consists of both dynamic (OBU) and static (RSU and HBU) type of nodes and changes the network topology continuously (Figure 4.1). An example scenario, vehicles (OBUs) are parked at owner's residence and form M2M social networks with respective HBUs. As both OBU and HBU are static in this scenario, we consider it as static SIoV, which extends to neighbouring OBUs and HBUs based on the communication area of an OBU. For example, all the cars parked in an apartment's basement or in the parking lot can form SIoV with HBUs of the building. In another scenario, one OBU leaves its residential HBU and after a travel arrives at a remote HBU. The visiting OBU in the contact with a remote HBU is in a temporary static relationship. Example: a car comes in contact with its mechanic's HBU. The third scenario is highly dynamic, where an OBU is moving on the roadway and forms wireless networks with other OBUs and RSUs to exchange safety and non-safety services. These wireless relations are very temporary, and all the mentioned relationships are represented as virtual links in the SIoV data social graph.

We also adopt the social relationships described in SIoT [37], where Parental Object Relationship (POR) exists between a vehicle, and its manufacturer and manufacturer has

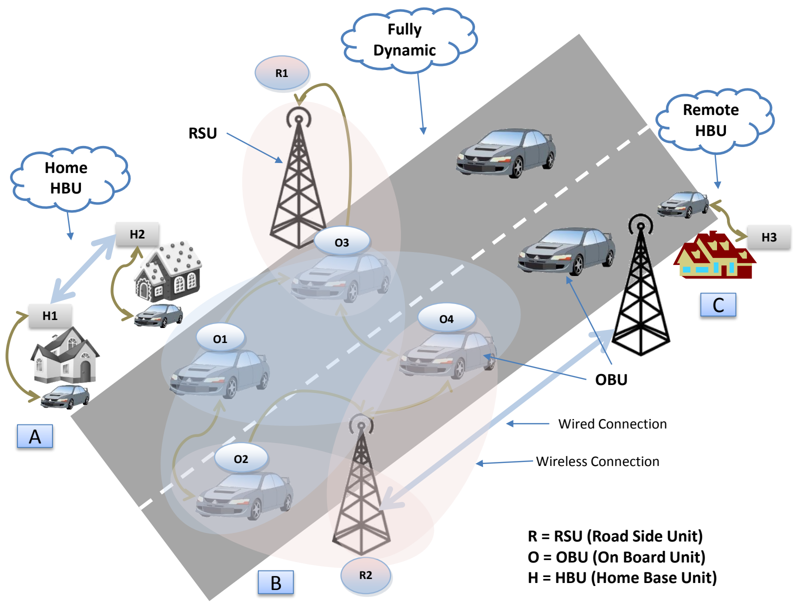


Figure 4.1: Social network structures: A) Vehicle's home based static social network, B) Fully dynamic social network, C) Visiting vehicle's dynamic social network

the initial responsibility to enable the vehicle with public settings for SIoV. Manufacturers can introduce new features to their vehicles and add custom sections in a *tNote* [70] message. Co-Location Object Relationship (CLOR) applies when two objects are working close in a geographical location. Whereas Co-Work Object Relationship (CWOR) applies when two objects are working together to achieve a goal regardless of their location. Both CLOR and CWOR apply to OBU-OBU type communication, where vehicles work together situated in close geo-locations. Ownership Object Relationship (OOR) represents the OBU-HBU(resident) relationship, where an owner has the authority to configure the vehicle's private settings. Again, Social Object Relationship (SOR) applies to OBU-HBU(remote) when a vehicle's owner is willing to share protected information with friends (e.g. vehicle status with the car mechanic). We also have defined an additional relationship named Guardian Object Relationship (GOR) in order to define the communication of OBU-RSU where OBU is a child node of the RSU super node and RSU changes over time on the roadway. All these relationships enable SIoV to apply different behavioral governing rules at different scenarios.

Interactions

OBU-OBU: When a vehicle (i.e. OBU) comes in contact with another vehicle, then, based on physical layer message exchange, a virtual link of type CLOR-CWOR is created between the communicating vehicles. This virtual connection and corresponding physical messages are stored in the OBU's storage. Every time a new vehicle comes in contact with a running vehicle, they exchange messages and store them in their social graph. Only public information is exchanged in this communication. But, private information is also kept stored in the social graph. This OBU-OBU social graph continues to grow in every OBU. If an OBU has acted as a platoon leader of an OBU-OBU network, then once it comes in the range of an RSU it transfers the bulk OBU-OBU social graph to the RSU. Only public information gets transferred in such cases.

OBU-RSU: When an RSU receives OBU-OBU graph data of interaction messages, the GOR transaction is completed. At this level of physical communication, a new virtual social link is created between the participating OBU and RSU. After this step, the network takes a shape where a group of OBU nodes form a small social network with a super node RSU. Every RSU, based on its wireless technology and GPS location, maintains a radius of geo-social space. It is possible to uniquely identify a specific message, hence redundancy is reduced in the OBU as well as in RSU.

RSU-RSU: Geo-locally neighbour RSUs generally have direct wired connection, which is a CLOR-CWOR relationship and is represented with a virtual link in the SIoV. If an OBU fails to complete the GOR transaction to the RSU, then the rest of the transaction is completed in the next RSU, and the neighbouring RSUs exchange the parts to complete the transaction.

OBU-HBU: When a new vehicle is produced, it is part of the manufacturer's plant HBU. So far, the owner of the vehicle is its manufacturer. A manufacturer is allowed to change the vehicle settings (e.g. choosing public message parts), which belongs to POR relationship. When the vehicle ownership belongs to a vehicle user, owner can change the non-public settings of the vehicle message which is part of OOR relationship. SOR relationship applies to OBU-HBU(remote) type of communication.

HBU-HBU: HBUs are connected through the Internet. They know their geographical locations as well and are in a CWOR-CLOR relationship and can exchange vehicular usage information among them.

Above discussions only focus on the physical level of communication in the VCPS. All the above relationships and interactions are also possible through the cloud based cyber P2P relationship layers as well. In this case, instead of ad-hoc networks, SIoV subsystem (e.g. vehicle) communicates with the cloud using the cellular networks. In case of C2PS, relationships among the *things* are managed by either the owners or authorities. But in case of VCPS, some relationships (e.g. V2V, V2I) are created for a short moment only and without any human interventions. To solve this issue, we consider that for every SIoV setup, there exists a start network smart thing, *Command Center*, that creates the cyber layer P2P connections (e.g. V2V, V2I) based on its knowledge about the physical things' (i.e. vehicles, infrastructures) geo-locations and virtual communication range.

4.1.2 SIoV System Components

In this section, we describe the component architecture of SIoV (Figure 3.13) that consists of six components: *tNote* Message, On Board Unit (OBU), Road Side Unit (RSU), Home Base Unit (HBU), *tNote* Cloud, and System Interface. In order to enable cross domain IoT integration, key SIoV systems have been designed following the IoT Architecture Reference Model (ARM[1]) [91][92] which promotes a common understanding between the research community and the industry to provide interoperable solutions in the communication level, service level, as well as across platforms. We adhere to the following guidelines given in

[1] Final architectural reference model for the IoT v3.0, http://www.iot-a.eu/public/public-documents/d1.5/at_download/file

IoT-A reference model while designing domain models for the SIoV systems:

- Categorizing abstract concepts as Device, Service, Resource, User or Physical Entity.
- Defining concepts using UML notations.
- Using color codes to represent abstract concepts.
- Active Digital Artifacts are running software applications, agents or services.
- Passive Digital Artifacts are database entries or digital representations.
- Every Physical Entity is represented by its corresponding Virtual Entity.
- Service exposes a functionality of a Device through its hosted Resources.

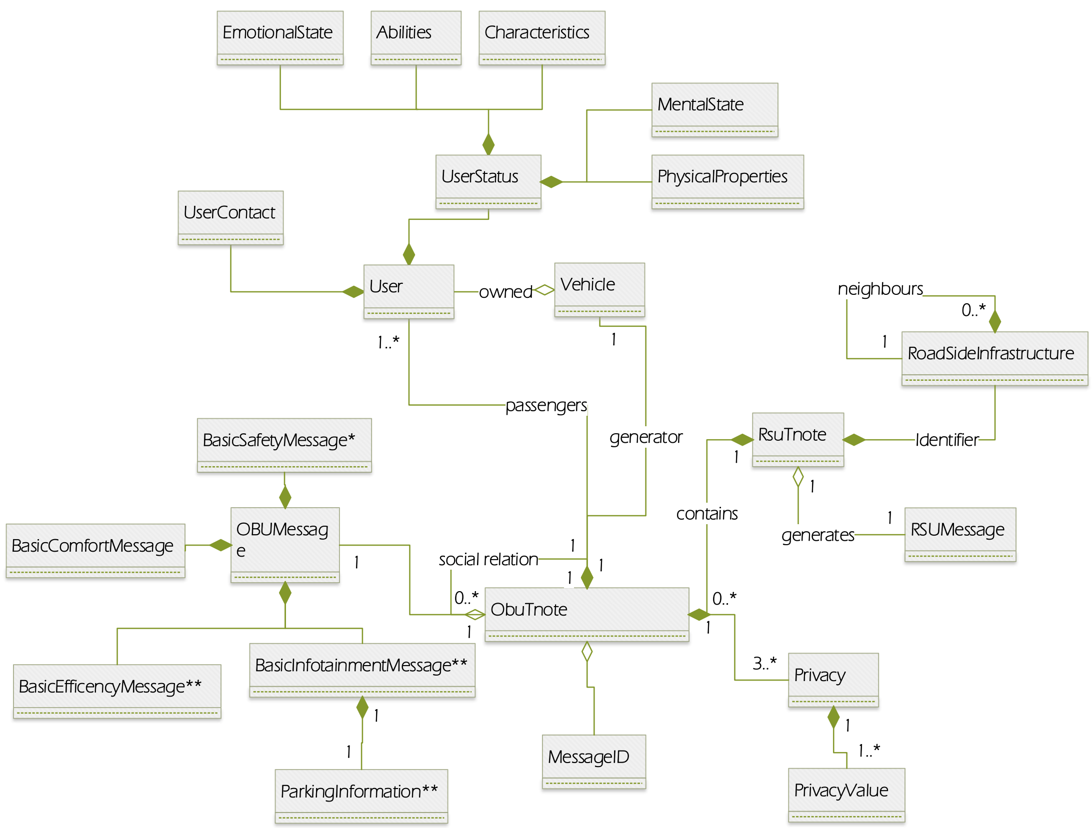


Figure 4.2: Information relation of the tNote message components

Following section describes the SIoV communications from the physical layer communication perspective. As described in the previous section, in a C2PS architecture, all the physical level communications can also be abstracted to occur through P2P based virtual connections of the cyber processes representing some physical layer systems. In this case, instead of using ad-hoc networks for data dissemination, physical things choose to use the cellular networks.

tNote **Message**

tNote message (Figure 4.2) is a wrapper over user information, vehicular status, and different types of sensory messages, which was introduced in [70]. The user part of the message contains both static administrative user information and dynamic user information such as physiological state, mental state, etc. The static part of the vehicular message contains information such as driver's identity, vehicle's physical attributes, interior, and exterior information. Whereas, the dynamic part of the vehicular message consists of the sensory messages and observed events. Different layers of privacy are maintained by dividing data into the public, private and protected categories. Details of the *tNote* message structure is covered in Section 4.2.

On Board Unit (OBU)

OBU plays the key role in sensing and building the vehicular interaction messages. Every vehicle is represented using a unique number such as IPV6, Universal Product Code (UPC), or Electronic Product Code (EPC). According to IoT-A, it should have a corresponding Virtual Entity (Figure 4.3). *Identity Manager* is responsible for the ID update management of OBU. Every vehicle hosts a list of On Board Diagnostic (OBD) sensor devices, and it can also integrate other internal sensory devices such as fatigue detector, sleep detector, etc. For every sensor, there is an on-device resource to monitor the activities. All these sensory data can be accessed through the services. A *Message Builder* accesses these data

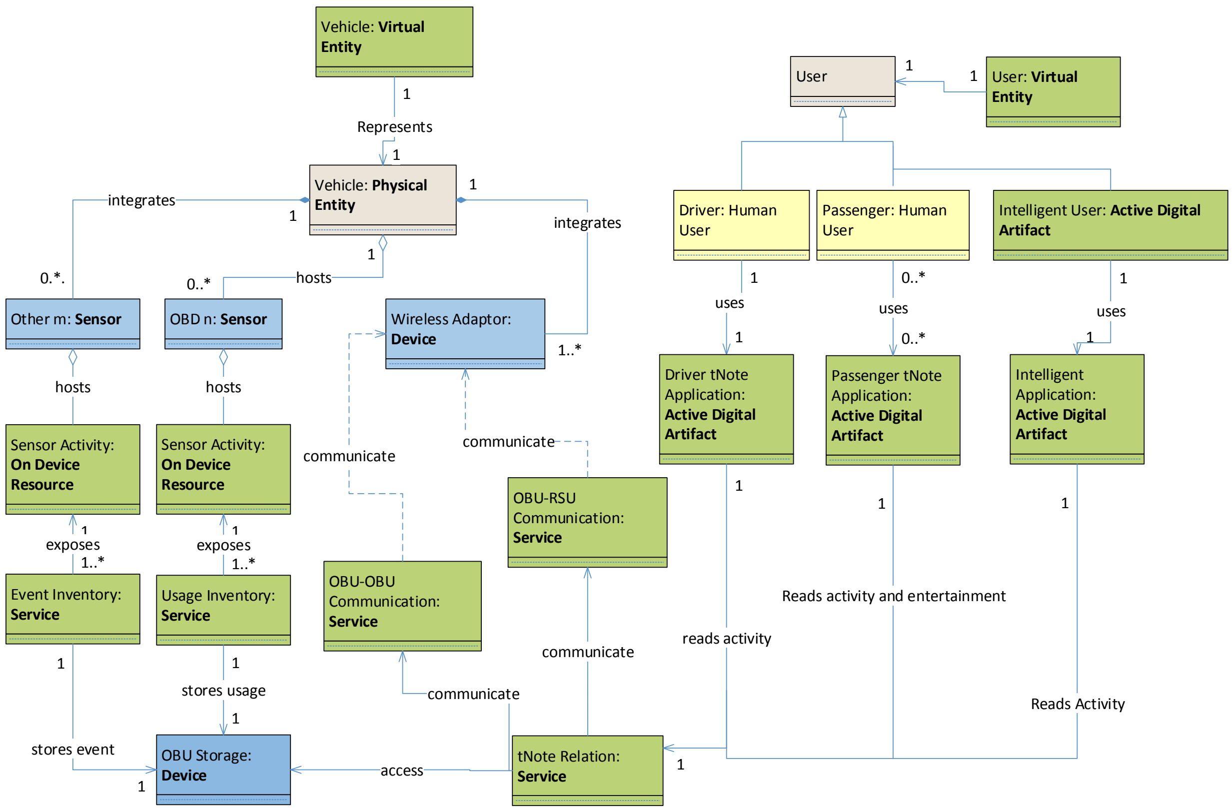


Figure 4.3: Domain model of On Board Unit (OBU)

to build a message following *Automotive Ontology* and stores it on the OBU storage. Every vehicle is expected to have at least one wireless adaptor to communicate with surrounding OBUs and RSUs. When an OBU comes in contact with another OBU, it exchanges public messages using OBU-OBU communication service through the wireless adaptor.

In an OBU-OBU ad-hoc vehicle platoon, there exists a platoon leader, which receives all the communication messages and later stores them in the OBU-OBU social graph with the help of *Data Manager*. Since OBU-OBU communication is omnidirectional, it is possible that messages are duplicated in the same OBU-OBU topology from different intermediate sources. *Data Manager* filters out stale messages and keeps the local OBU data structure up-to-date. Once the platoon leader OBU reaches into the range of an RSU, *Dispatcher* pushes the OBU based *tNote* data to the RSU using OBU-RSU communication service. All these data are consumed either by the vehicle operator, on-board passengers or intelligent software agents using respective application clients.

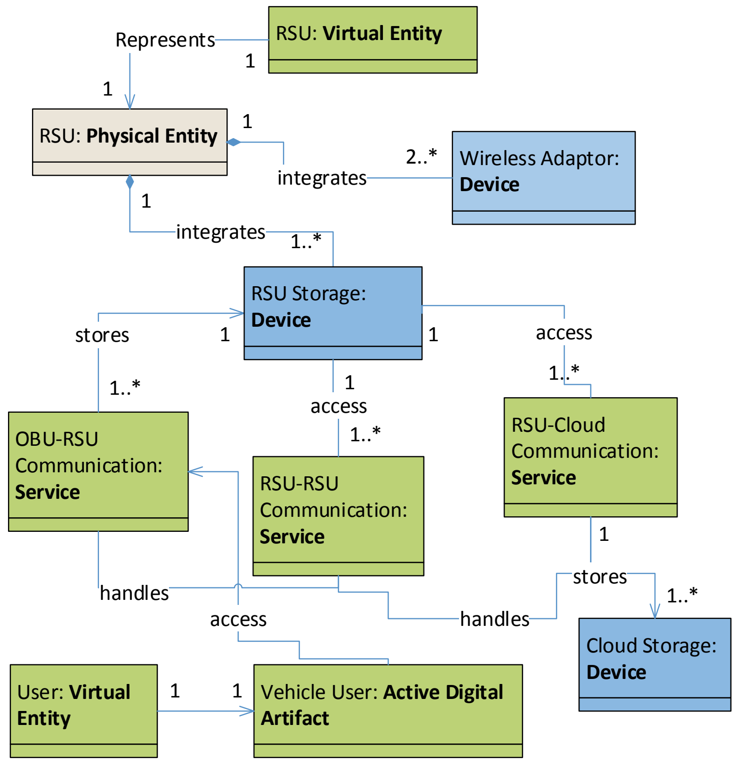


Figure 4.4: Domain model of Road Side Unit (RSU)

Road Side Unit (RSU)

In the SIoV, whenever an OBU comes within the range of an RSU, RSU asks the OBU, whether it wants to share the OBU-OBU social graph. If an OBU acted as a platoon leader in between the last RSU and the current RSU, then it should have some OBU-OBU social graph to push to the RSU. The *Identity Manager* of an RSU maintains the physical and virtual identity of the RSU entities. Every RSU has its geographical location, storage device and at least two communication interfaces. OBU-RSU communication occurs through the wireless network interface and uses the OBU-RSU Communication Service. At any specific time, one RSU can receive multiple *tNote* bulk messages from various approaching platoon leaders. RSU collects the *tNote* bulks, and *Social Tag Manager* assigns possible tags to the collected data before storing them to the cloud through RSU-Cloud Communication Service (Figure 4.4). In case an OBU-RSU data exchange renders incomplete, then RSU-RSU Communication Service is used to handle incomplete social data transactions with neighbouring RSUs. Social tags generated from the abstract concepts of the ontology facilitate in searching the *tNote* cloud. *Data Manager* enhances the OBU-RSU social graph by reducing the redundant data. RSU is the super node of the OBU-OBU social network in the *tNote* cload. After a predetermined period, the public type RSU-OBU social graph is transferred to the *tNote* data cloud by the *Dispatcher*.

Home Base Unit (HBU)

HBU plays an important role in the static network building of the SIoV. The home/remote social networks are built based on the data sent from the HBU. Every HBU has an *Identity Manager* to maintain the physical and virtual identity of the component. Geo-local neighbouring OBUs are connected to the SIoV through the super node HBU. It is assumed that the home base unit has a storage device where all the social data are stored temporarily (Figure 4.5). HBU based user data and the corresponding relationship are managed by the *Data Manager*. Every HBU is expected to have network interfaces to exchange data with the OBU and the *tNote* cloud. The owner of the vehicle can change the privacy settings of the vehicle through the *tNote* Settings Service. *Privacy Settings* helps in privacy management of all devices connected to the HBU except the vehicles. All the private information of the OBU is transferred to the cloud through HBU. OBU-HBU Communication Service coordinates the functionality of other services: HBU-HBU service for inter-HBU data exchange, and HBU-Cloud service for data transferring to the central cloud.

***tNote* Cloud**

Data cloud is the central infrastructure that retains all the vehicular interactions (i.e. OBU-OBU, OBU-RSU, OBU-HBU) data along with their timestamps. It is hosted in the Internet and allows offline access to the content. *Topology Optimizer* runs on the offline data to remove existing redundancy. Ontology based formatting of the data enables

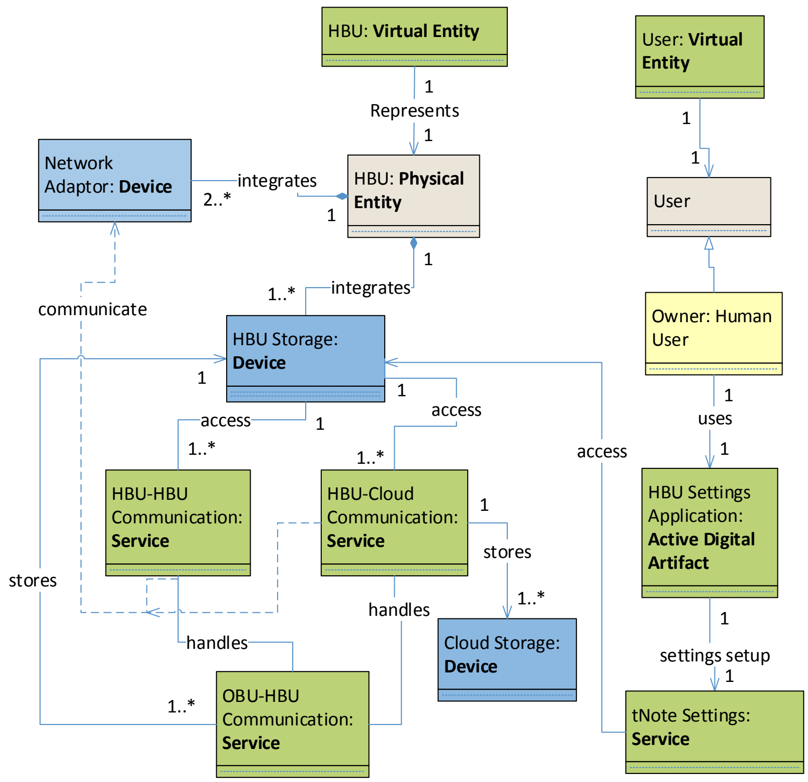


Figure 4.5: Domain model of Home Base Unit (HBU)

reasoning systems to provide additional insights based on new rules. This is a key aspect for *Intelligent Active Digital Artifacts* (Figure 4.6). All the queries sent to the cloud from different types of users access the *tNote Relation Service* and finally processed by *Query Processor*. Users of cloud can be intelligent software agents or human users. Human users can be of two types: social network user and transport authority. Social network users can create different interest groups (e.g. based on car model, common routes), update vehicle profiles, analyse their usual route data, analyse future travel plans, etc. using the cloud data. Whereas transport authorities can use the system from BigData perspective and develop various mining applications to solve transport related problems.

System Interface

We describe this section from user perspective, where the system interface can be divided into *Profile, Routes, Friends, Groups, News Feed, Subscription* and *Social Graph*. The

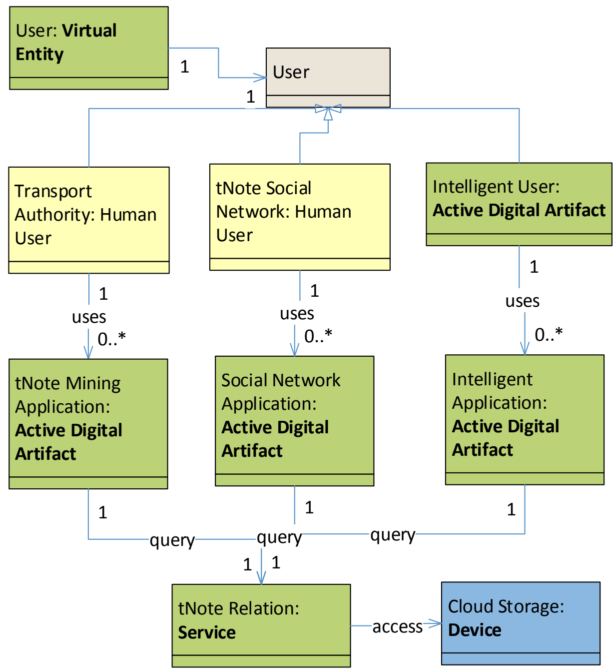


Figure 4.6: Domain model of *tNote* Cloud

Profile presents all the up-to-date vehicle related public, private or protected information. All the public information about the vehicle is easily accessible in its social graph. But, private information is only available for the owner of the vehicle. *Social Graph* of a vehicle is the node-link relationships among OBUs, RSUs, and HBUs involved with that vehicle. The social graph of any vehicle represents the friend structure of any vehicle at a time. *Friends* of a vehicle are different than the friends of a person. Vehicular friends represent the neighbours which participated in the message exchange at any specific time. *Routes* collects all the frequently travelled routes. *Groups* collect different interest groups such as based on owner's interest, manufacturer, vendor, travel interest, etc. Any vehicle or user can subscribe to the important updates of a friendly OBU or RSUs. Updated data are accessible through the *News Feed* like web interface.

4.1.3 Privacy and Security

Privacy is maintained in SIoV by dividing content types to: private, protected, and public. Private type of information is only accessible by the owner of the vehicle (e.g. user's

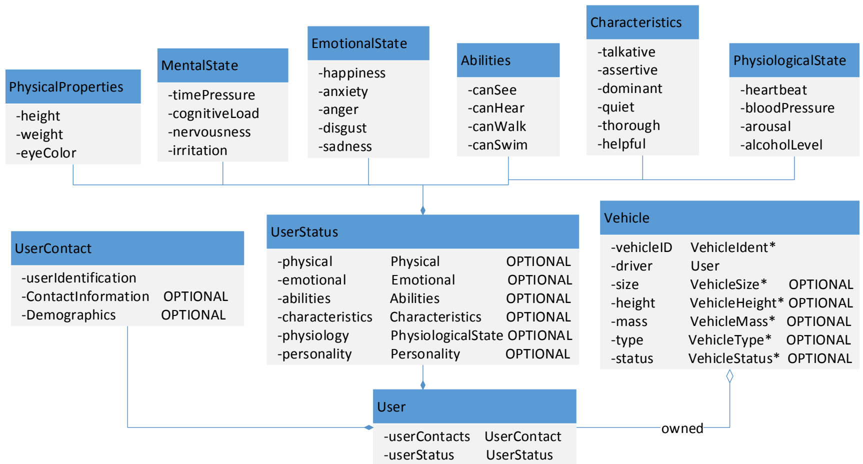


Figure 4.7: User and Vehicle components of the *tNote* message. Here * comes from SAE J2735.

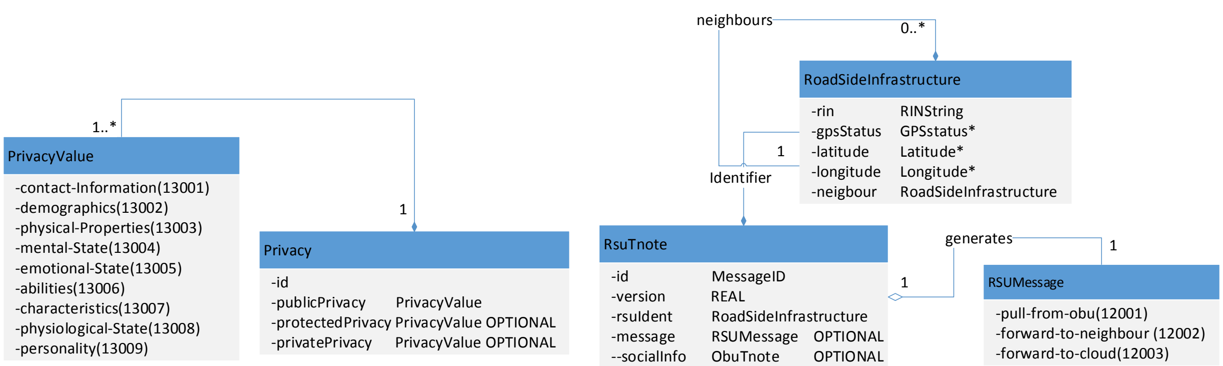


Figure 4.8: RSU and Privacy components of the *tNote* message. Here * comes from SAE J2735

personal information, vehicle usage data). Any part of the *tNote* message not determined public is by default private. Owner has the authority to share the private information to the public or to a selected group by making them protected (e.g. vehicle usage data shared with the mechanic). A national or international body (e.g. International Telecommunication Union (ITU)) can decide which part of the message must be public (e.g. Basic Safety Message) that is accessible by anyone (i.e. vehicle, user, authority) participating in a social communication. The sensory data generated in each OBU is stored in various parts of the *tNote* message structure. Only public content is shared between the OBUs, later OBU cluster data are gathered in the RSUs, and finally most important parts are transferred to the cloud. All the private, protected types of data are shared to the cloud through the

HBU of a user.

VCPS is mostly data centric and can be considered as an application on the IoT infrastructure. So, SIoV follows all the security measures implemented in the VANETs by different service providers such as authentication, access control, or secured service access to protect the network against all possible types of attacks such as internal, external, active, passive etc. [93][94]

4.2 tNote Message Structure

In this section, we describe the detailed structure of the *tNote* message. In order to define the message structure, we consult to automotive ontologies such as [95][96][97], SAE J2735 Dedicated Short Range Communications (DSRC[2]) Message Set, and `ITISEventType`[3] of Advanced Traveler Information System (ATIS) schema. Every message is transferred as one unit in the communication channel. The *tNote* message works as a wrapper over the other messages. Details of the message parts are described below.

4.2.1 User and Vehicle

Figure 4.7 shows the detailed relationship of user status and vehicle in a *tNote* message. User status consists of physical properties, mental state, emotional state, abilities and characteristics. It is assumed that the sensors inside a smart vehicle will be able to detect and notify these user states. At any given time, abilities, emotional state and mental state are the most important factors for an operator's driving capability. These *tNote* values will help the intelligent vehicle system to be aware about their surrounding vehicle operators' contexts. For the `UserStatus`, most of the corresponding values are `OPTIONAL`, i.e., those values are used when required. `OPTIONAL` values in the message structure ensure flexibility

[2] http://www.sae.org/standardsdev/dsrc/

[3] http://www.itsware.net/ITSschemas/ATIS/ATIS-03-00-79/OxDocs/ITIS-Adopted-03-00-02.xsd.html

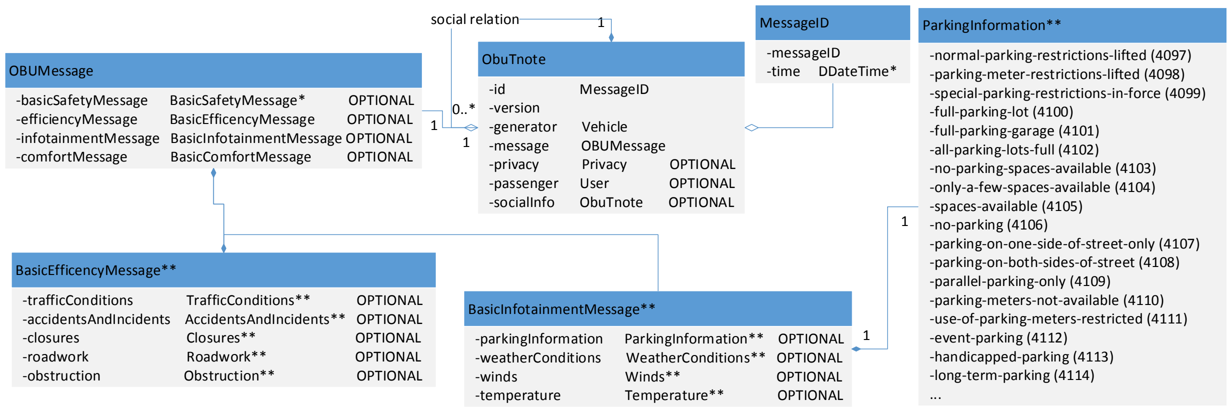


Figure 4.9: OBU components of the *tNote* message. Here * comes from SAE J2735 and ** comes from ATIS schema

as well as strong handles to reduce the payload if required. `Vehicle` is mostly derived from the SAE J2735 message set. Here `VehicleIdent` refers to the vehicle identification class of SAE J2735 which is consist of VIN, owner code, vehicle type, vehicle class, etc. Vehicle class also contains information such as vehicle height, mass, status, etc. `VehicleStatus` provides related information about lights, wipers, brake, steering, acceleration, speed, gps, etc.

4.2.2 Privacy

Privacy is an important part of the *tNote* message. From Figure 4.8, we see that *Privacy-Value* can be easily extended and represented in binary form, and each *tNote* message can mention which part of the message belongs to what privacy: private/protected/public. As mentioned before, public information is shared in the OBU-OBU and OBU-RSU communication. Private and protected information is shared through the OBU-HBU communication to the cloud.

4.2.3 RSU Message

Every *tNote* message is generated in the vehicle and at some point transferred to its closest RSU. An RSU is expected to receive bulk *tNote* messages from platoon leader OBUs. Bulk message consists of *tNote* messages structured in a tree format, which represents the OBU-OBU social graph. In Figure 4.8, we find that `RsuTnote` consists of a list of `ObuTnote` messages. Every RSU is represented using an identifier such as `RoadSideInfrastructure` which recognizes its neighbours and the geo-location. The job of RSU is to pull data from OBU, forward incomplete transactions to neighbours, and forward complete transactions to the *tNote Cloud* as described in `RSUMessage`. These functions can easily be extended by adding new RSU functionalities.

4.2.4 OBU Message

An OBU generated message, `ObuTnote`, can be composed of message id, information about the vehicle that created it, privacy segment list, original message part and list of social information in a tree structure. Original message contains the safety, efficiency, infotainment and comfort related metadata (Figure 4.9). Most of the `BasicSafetyMessage` message `Part I` are only transmitted in the VANETs control channel and few selected `Part I` and important `Part II` events are stored in the platoon leader OBU, wrapped in the `ObuTnote`. Other types of messages are also stored in the platoon leader OBU. `socialInfo` is a tree like data structure to store the `ObuTnote`s received from the neighbours. `BasicSafetyMessage` is collected from the SAE J2735 message set. *BasicEfficiencyMessage* is a special case of safety message that cannot be described in the original safety message and `BasicInfotainmentMessage` is derived from the Advanced Traveller Information System schema. As can be seen from the `ParkingInformation`, all these messages are already defined in the ATIS schema. We can select the important fields for infotainment applications or add them to our need. `BasicComfortMessage` is applicable for video or audio media shared among the OBUs while on the roadway [71].

4.3 Summary

In this chapter, we describe the vehicular Cyber-Physical System, *SIoV*, using IoT system design philosophy. We have identified the M2M social interactions and relationships that are evident in the VCPS communications. Also, IoT foundation is instilled in the VCPS architecture by consulting to the standard IoT-Architecture reference model and by mapping existing VANETs systems to the IoT design philosophy. Additionally, we have proposed a wrapper message structure, *tNote*, to organize vehicle interior-exterior information, driver status, various safety or non-safety vehicle data along with M2M social relationships. For this purpose, we consult to automotive ontologies, SAE J2735, and ATIS schema. In the SIoV interactions, different levels of *tNote* data cloud are stored in the OBUs, RSUs, HBUs and to the cloud. Privacy aware organization of this social data can provide an easier handle for the data exchanges as well as easy search and intelligent access.

Chapter 5

Model-based Reconfiguration of VCPS

The new paradigm of computing, CPS, is receiving increased interests from the research community, since it perfectly coincides with the embedded systems, cloud computing, and service oriented architecture. The vehicular domain of CPS, VCPS, requires to handle some domain specific problems such as vehicle mobility and the resultant short life span of communication channel between moving vehicles. The data cloud, generated at different levels of VCPS communications, need to be properly managed so that the most useful information is stored. At the same time, based on the current setup of traffic parameters, VCPS subsystems need to readjust themselves to ensure overall VCPS service quality is ensured. To tackle this problem, it is important to have understanding of data workload on different VCPS subsystems.

In this chapter, we study the effects of data workloads on the proposed VCPS architecture, SIoV, at different traffic parameters and VCPS scenarios that were published in [73]. The models developed in this study are important to dynamically monitor and manage the overall data workloads on a VCPS setup. We validate the proposed workload models using extensive simulations and measure the effects on the workloads at different system settings. This allows us to find the relationships among the system parameters, which

Table 5.1: Symbols related to data workload model

Symbol	Description
v	Average speed of the vehicle
L	Distance between two Infrastructures
T	Time required for a vehicle to reach next I with v speed
s	Sensor sampling rate
λ	Data rate of information type
r_v, r_i	Communication range of vehicle and infrastructure
n_h, n_a	Number of home friends and number of homes
d	Sample size
D	Data workload
f	Data fraction function
i, j, k	Numbering parameters
$p(x)$	probability of a successful event x
e	Occurrence of an event
ψ	Storage requirements of a subsystem

are used to design example adaptive algorithms for the subsystems to adjust the overall system goal. We first design a basic VCPS scenario in the Section 5.1, which is further extended to adopt complex scenarios in Section 5.2. Example adaptive algorithms for the subsystems are discussed in Section 5.3.

5.1 Basic SIoV Data Workload Model

We take an incremental approach to model SIoV; first, we model the basic scenario, and then we gradually extend the basic model to incorporate more complex scenarios. The basic scenario (Figure 5.1) consists of a vehicle that belongs to a household and travels from the infrastructures I_k to I_{k+1} on a one way road. The vehicle offers services related to safety, efficiency, comfort, and infotainment. As there are no other vehicles offering their services, no sharing occurs; the sensory information generated by one vehicle is stored in its OBU-OBU social graph. When this vehicle reaches the infrastructure I_k, it hands over the data to it, and I_k builds OBU-RSU social graph. After a predefined time, the infrastructure transfers the OBU-RSU social graph to the cloud. Following sections describe the workload

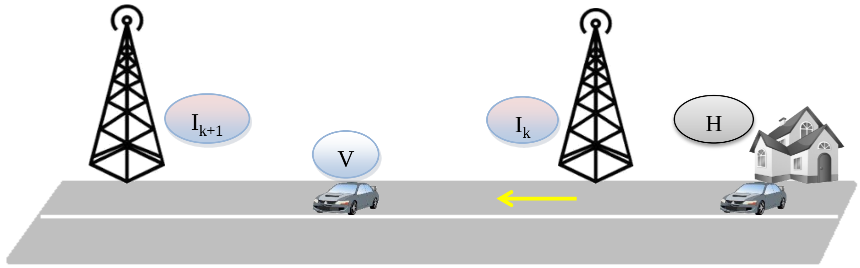


Figure 5.1: Scenario of the basic Social Internet of Vehicles (SIoV) model.

models for different SIoV system components in this setup. In the proposed system model, workload is measured in terms of data storage requirements for the SIoV subsystems in a time period T:

- Vehicle data storage (ψ_v) is used to store OBU-OBU social graph that is composed of sensory information and V2V interactions.
- Infrastructure data storage (ψ_I) is used to store OBU-RSU social graph that is generated in V2I or I2I interactions.
- Home data storage (ψ_h) is used to store incomplete V2I and H2H interaction data.
- Cloud data storage (ψ_c) is required to aggregate regional or residential infrastructure and home data.

5.1.1 Assumptions

We assume that the vehicle is travelling at a constant speed (v) in between the infrastructures that are L distance apart, there is no traffic jam, the traffic is one way, and each vehicle has r_v communication range. Every vehicle is equipped with basic and extended sensors. Basic sensors are used to generate public information for safe manoeuvring (e.g., VANET "i am here" message). On the other hand, extended sensors generate intermittent messages that describe events occurring in and around the vehicle (e.g., sharp brake, lost control, obstacles ahead, wrong direction, *etc.*) (Figure 5.2). In addition, messages also contain the interaction information (e.g., efficiency, comfort, entertainment requests) which is different from the requested content.

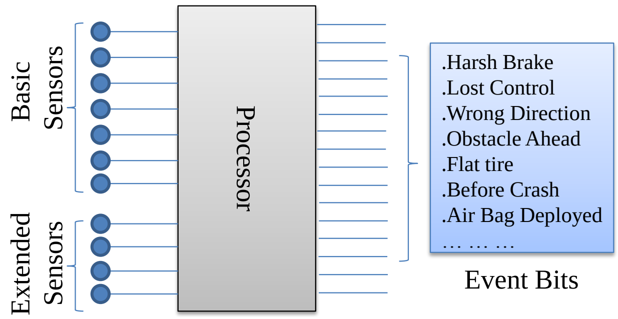


Figure 5.2: Sensory data to events identification.

5.1.2 Event Definition

Every type of service requests are considered as events. Events can also be detected by processing the sensory data. All the event occurrences are represented using a fixed bit and (on/off) value (Figure 5.2).

In case of regular "*i am here*" type message, only the basic sensory data are shared with the surrounding vehicles. When extended sensors detect an event, all the sensory data (*i.e.*, basic + extended) including representative event bits are also shared with the neighbouring vehicles either through physical connection or through virtual connection. In an event byte, a checked bit represents an event occurrence. Multiple checked bits in a byte represent multiple event detections. For example, 16 and 32 event bytes can represent individually 16×8 (bits) =128 and 32×8 (bits) = 256 events respectively.

5.1.3 Interaction Data

In the SIoV, the messages generated by vehicles consist of two parts, dynamic and static. Static part is of fixed size and includes vehicle or user related administrative information such as vehicle number or user details. Dynamic part consists of indoor and outdoor sensory data, and other social service requests related to comfort, efficiency, and entertainment. The service requests can use physical ad-hoc communication range, or use virtual communications if out of physical range through 3G/LTE type cellular networks.

If n_b and n_x are the number of basic and extended data sensors respectively deployed in a vehicle, s is the sampling rate of a sensor, and d is the sample size, then the vehicular sensory data generation rate can be

$$\text{Basic data rate, } \lambda_b = \sum_{i=1}^{n_b} (s_i * d_i) \tag{5.1}$$

$$\text{Extended data rate, } \lambda_x = \sum_{j=1}^{n_x} (s_j * d_j) \tag{5.2}$$

The static information about a vehicle rarely changes over the time such as vehicle identification (e.g., serial number, manufacturer, model), physical attributes (e.g., length, height, weight), vehicle exterior (body, axles, frame), *etc.* This information is synchronized to the cloud through home (H), when a new vehicle joins to the SIoV. This information represents both the physical and the virtual vehicle. If λ_{sv}, λ_{dv} are the data rate of static and dynamic vehicular information respectively then total vehicular data rate $\lambda_v = \lambda_{sv} + \lambda_{dv}$ where $\lambda_{dv} = \lambda_b + \lambda_x$.

Again, user descriptive message has two parts: (1) static information such as name, ID, license number, age, *etc.* (2) dynamic information that changes over time such as physiological parameters and mental state obtained through sensors [72]. A new user (*i.e.*, driver) of SIoV shares his private static information (λ_{su}, data rate) through the home. Whereas, the public dynamic user information (λ_{du}, data rate) is shared with other vehicles and infrastructures. Here, $\lambda_u = \lambda_{su} + \lambda_{du}$.

If du_e is the number of dynamic user status entities, n_e^u is the number of properties in an entity, d_{ij} is the size of ith entity's jth property, and s_{du} is user status sampling rate, then λ_{du} can be

$$\lambda_{du} = s_{du} * \sum_{i=1}^{du_e} \sum_{j=1}^{n_e^u} d_{ij} \tag{5.3}$$

Here, λ_{su} and λ_{sv} contributes to static type data and λ_{du} and λ_{dv} produces dynamic data. Also, other dynamic data requests are generated from efficiency, comfort and entertainment type requests whose data rates are λ_{ef}, λ_{cm}, and λ_{en} respectively.

5.1.4 Data Workload Measures

We consider two types of interactions in the SIoV: push and pull. In the push type interactions, data is shared without any specific request from others, such as safety information sharing. These interactions are given highest priority and data can travel in any path physical or virtual whichever way is the most efficient. On the other hand, pull type interactions take place in response to the requests from others, such as request for traffic status, parking information or entertainment media sharing, *etc*. The type of the interactions are used to describe the actions of the system, they do not have effect on the model design. Following sections describe the data workload measures of the basic scenario case.

Vehicle Data Storage (ψ_v)

We consider the time a vehicle takes to travel from infrastructure I_k to I_{k+1} as T and all the successful interactions (e.g., V2H, V2V, V2I) as vehicle related events. Figure 5.3 shows how different types of interactions take place over T time. Basic safety information is continuously shared with a data rate of λ_b. Extended safety information (e.g., Bumpy road) is only shared during few time intervals with an additional data rate of λ_x. Also, the dynamic user information can be shared intermittently at λ_{du} rate. Similarly, efficiency, comfort and entertainment message requests incur λ_{ef}, λ_{cm} and λ_{en} data rates respectively during different time intervals. A vehicle stores all the data generated by itself (push interactions) and data received during social (pull) interactions. If D_g indicates the data generated and D_r represents the data received, then the total data of any subsystem is $D = D_g + D_r$. The data storage required (ψ^v) by a vehicle while travelling between two infrastructures and only generating sensory observations (not receiving) can be represented

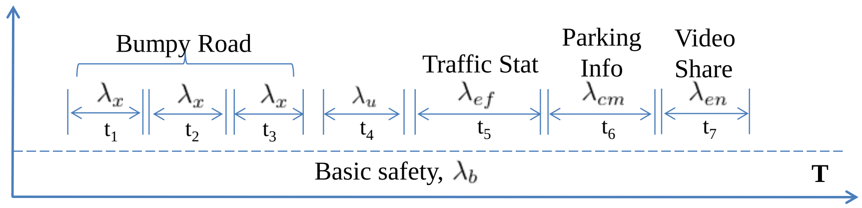


Figure 5.3: An example timeline of vehicle storage, ψ_v in T time.

as,

$$\psi_v = \lambda_b * T + \sum_{i=1}^{n_e} (\lambda_i * t_i), \text{ where, } \lambda_i \in \{\lambda_x, \lambda_{du}, \lambda_{ef}, \lambda_{cm}, \lambda_{en}\} \tag{5.4}$$

Here n_e is the total number of interactions, λ_i is the data rate corresponding to ith interaction type; and t_i is total duration of ith type of interaction. The time required for each type of event can be different based on the communication range and vehicle speed. Safety like push based data generation is considered mandatory and other pull based interactions are optional for the provider vehicle. Note that in the basic scenario, where there is only one vehicle, there are no pull type interactions ($i.e.$, $\lambda_{ef} = \lambda_{cm} = \lambda_{en} = 0$). Also, in this model we are only considering dynamic information. Because, the overhead of storing static information (λ_{sv} and λ_{su}) is negligible in comparison to the dynamic data.

Infrastructure Data Storage (ψ_I)

In the basic scenario, we have only one vehicle passing through the infrastructures. Hence, it is part of a platoon of one member and all the data generated while travelling will be transferred to the infrastructure (I_k). As mentioned earlier, if the vehicle is unable to transfer all the data, then the remaining data is transferred to the next infrastructure in line (e.g., I_{k+1}). The next infrastructure sends the data back to the originating infrastructure through I2I interaction. In addition, the infrastructure may also receive other types of I2I interaction requests (Figure 5.4) from the neighbour infrastructure. If we consider the interactions in T time then ψ_I^k can be represented as,

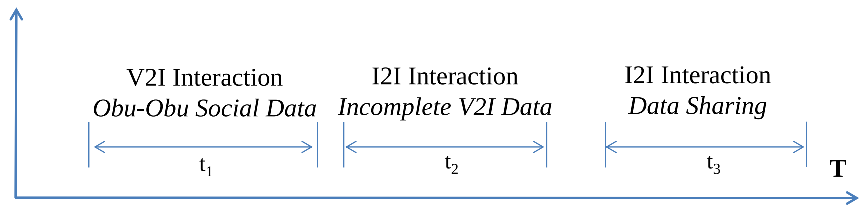


Figure 5.4: An example timeline of infrastructure storage, ψ_I in T time.

$$\psi_I^k = \psi_v + f_I(\psi_I^{k-1}, \psi_I^{k+1}) \tag{5.5}$$

where ψ_v is the data contributed by a vehicle and f_I returns the shareable data of the right $(k+1)$ and left $(k-1)$ neighbours of any infrastructure. The number of neighbours can be more than two based on the shape of the road layout and how many different traffic directions approaching the infrastructure.

Home Data Storage (ψ_h)

Every vehicle belongs to a household where it shares its private information or uncommitted V2I or V2V data using V2H interactions. Static user data ($\lambda_{su} * t_2$) and static vehicular data ($\lambda_{sv} * t_1$) are also shared through the V2H interactions where t_1, t_2 are the time to complete respective transactions. Moreover, services can be shared with the local or remote friends using H2H interactions. But in basic model we consider only one house, hence the data storage requirements for the home is

$$\psi_h = f_h(\psi_v) \tag{5.6}$$

Here f_h is the function that provides the uncommitted vehicular data fragment of the residence vehicle.

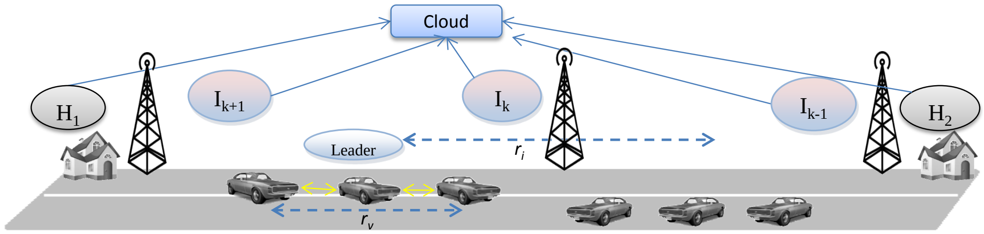


Figure 5.5: A scenario of the extended model.

Cloud Data Storage (ψ_c)

All the data collected in the infrastructure and the home are committed to the cloud. Hence, for the basic model of SIoV, cloud storage requirements can be depicted as

$$\psi_c = \psi_I + \psi_h \tag{5.7}$$

5.2 Extension of Basic Data Workload Model

The basic model describes the SIoV scenario where one vehicle belongs to a household and travels from one infrastructure to the next infrastructure. We further extend the basic model to, (1) incorporate more vehicles to the dynamic scenario so that the vehicles can form platoons of multiple members, 2) introduce a region with multiple infrastructures so that inter-infrastructure communication can be logged, (3) add more houses to the residential areas to extend the number of local and remote friends (Figure 5.5). Following sections describe the models in detail.

5.2.1 Multiple Vehicles Scenario

In the multiple vehicles scenario, while moving between the infrastructures, a vehicle becomes part of multi-member platoons (Figure 5.6) over time based on its communication range. In a multi-member platoon, all the vehicles generate (D_g) their sensory observations as well as share (D_r, V2V interaction) them with the other members. So, the number of vehicles interacting at any time can be represented as $n_v(t)$. In the practical scenario, $n_v(t)$

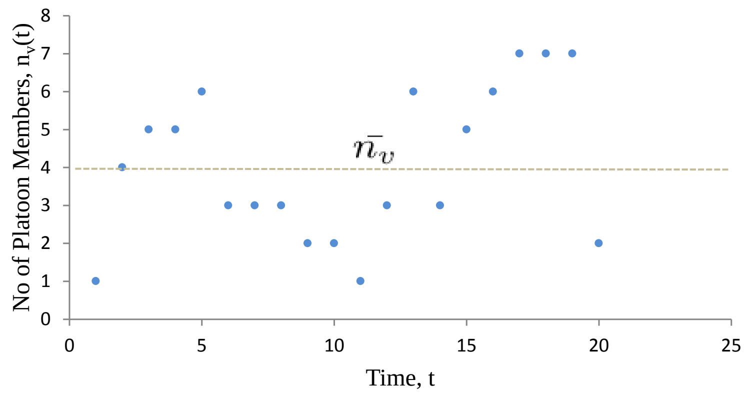


Figure 5.6: In the extended model, a vehicle becomes member of multi-member platoons over time and receives sensory data from many neighbouring vehicles.

depends on the traffic arrival rate, vehicle speed, traffic density and the communication range. The data generated and received by any vehicle in a multi-vehicle scenario can be represented as,

$$\psi'_v = D_g + D_r \rightarrow \psi_v + \frac{\psi_v}{T} * \int_0^T (n_v(t) - 1)\, dt \rightarrow \psi_v * \bar{n_v}, \text{ where, } \bar{n_v} = \tfrac{1}{T} * \int_0^T n_v(t)\, dt \quad (5.8)$$

Here, $\bar{n_v}$ is the average number of vehicles that are involved in message sharing at any time. Equation (5.8) shows that the number of vehicles in a platoon can change at every time step. Here, the connection life of any platoon depends on the communication range and the speed of the vehicles. If t_p is the average time for which each platoon connectivity lasts then $n_p = \frac{T}{t_p}$ is the number of platoons available in between two infrastructures.

Since the interaction can go through the physical layer or the cyber layer and wireless communication is omnidirectional, so we can assume a vehicle will receive most of the messages delivered in the platoon. If one vehicle is selected as a platoon leader at every time, then the platoon leader can only share the data with the infrastructure.

5.2.2 Multiple Infrastructures Scenario

After introducing more infrastructures to the last scenario, the existing model gets extended to a region of infrastructures (Figure 5.7). In this case, we have more V2I and

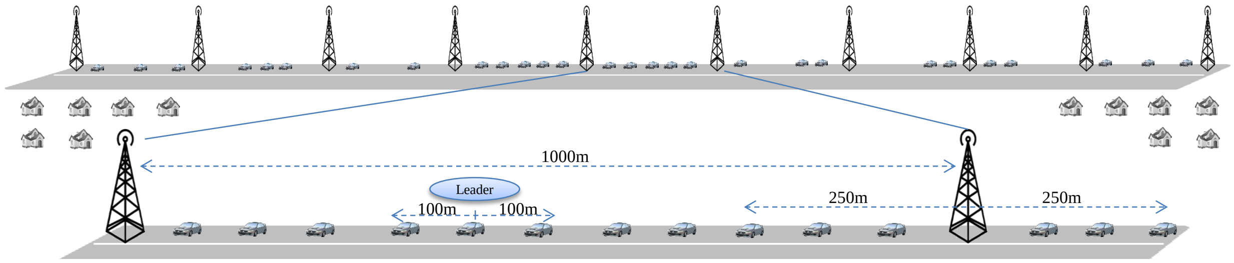


Figure 5.7: A detailed scenario setup of the extended model.

I2I interactions in addition to more V2V interactions. In case of V2I interactions, once a vehicle reaches the range of an infrastructure, it uses *push* method to share the OBU-OBU social graph. We assume the number of vehicles approaching an infrastructure in the one-way scenario per unit of time (e.g., a second) is a Poisson process with mean $\frac{1}{\rho}$ [62]; each infrastructure has a communication range of radius r_i; and it is operating for T time. So the workload of an infrastructure in T time can be regarded as

$$\psi_I'^k = \rho * T * \{\psi_v' + f_I'(\psi_I'^{(k-1)}, \psi_I'^{(k+1)})\} \tag{5.9}$$

Here, I_k can receive the incomplete V2I exchange data from I_{k-1} and other request can come from both I_{k-1} and I_{k+1} through I2I interactions. f_I' is a function whose parameters are all the infrastructures that are directly connected to kth one and an algorithm decides how much data of each neighbour will be shared. The algorithm will adapt to scenarios with two way traffic as well as different road layouts. In this model, we are considering each infrastructure has two neighbours and they are placed in a string. In a multi-way traffic, the workload can be multiplied by the number of vehicular directions approaching, $\psi_I''^k = n_d * \psi_I' k$, where n_d is the number of directions [62].

With more infrastructures, we get a dynamic geographical region where a string of infrastructures are placed and vehicle platoons move from one block to the next infrastructure block. Lets assume, there are n_I number of infrastructures in a region, hence the

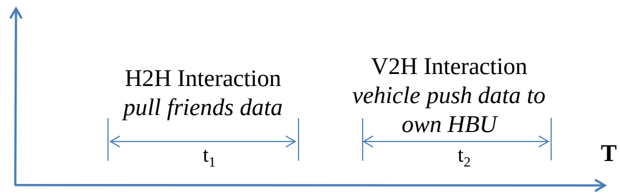


Figure 5.8: Event timeline of Home Base Unit (HBU) storage, ψ_h in T time.

workload of a particular region in T time is

$$\psi_{reg}^{T} = \sum_{k=1}^{n_I} \psi_I^{\prime k} \tag{5.10}$$

5.2.3 Multiple Homes Scenario

In case of multiple homes scenario, static friends regions grow, which offer greater level of V2H and H2H interactions. In the V2H interaction, one vehicle uses `push` to send usage data to its corresponding HBU storage (ψ^h). Whereas in H2H interactions, a home can share data with static friends using `pull` request (Figure 5.8). We assume that a home stores a fraction of the data which is collected by a vehicle in T time, $p(e_h)$ is the probability of a successful ($i.e.$, a `pull` request is answered) H2H interaction and n_h is the number of requests a home makes or receives. Then the workload of a house can be represented as

$$\psi_h' = f_h(\psi_v') + p(e_h) * \sum_{k=1}^{n_h} f_h'(\psi_v^{\prime k}) \rightarrow f_h(\psi_v') + p(e_h) * \sum_{k=1}^{n_h} f_h'(\psi_v^{\prime k}) \tag{5.11}$$

Here, f_h and f_h' deliver the data fragments of the vehicle and interaction with static friends respectively. If n_a is the total number of homes available in a SIoV instance then the residential workload can be regarded as

$$\psi_{res}^{T} = \sum_{k=1}^{n_a} \psi_h^{\prime k} \tag{5.12}$$

5.2.4 Central Data Cloud Workload

All the data that are collected in the home HBUs or infrastructure RSUs are transferred to the central social-graph cloud after a certain time. Home operation time is selected by the owner of a vehicle and the infrastructure time is selected by the SIoV provider of a region. For a region of infrastructures and residents, the total workload of the SIoV cloud in T time can be represented as

$$\psi_c^T = \psi_{reg}^T + \psi_{res}^T \tag{5.13}$$

If the time period is t_1 to t_2 then the cloud workload can be

$$\psi_c^{t_1,t_2} = \frac{(t_2 - t_1)}{T} * (\psi_{reg}^T + \psi_{res}^T) \tag{5.14}$$

5.3 Dynamic Adaptation of the SIoV Subsystems

Characteristics analysis from the Section 6.2.2 provides the basis for SIoV parameters' relationships, which are further used to develop the adaptation strategies for individual SIoV subsystems. In SIoV, every physical component is aware of its cyber entity by one-to-one mapping. As a result, any adaptation in the physical layer is notified to the cyber layer and *vice versa*. While designing the adaptation strategies, preferences are given to the locally known parameters over the global ones for every subsystem. Main objective of the adaptation techniques is to collect as much sensory data as possible while keeping the overall data workload to an acceptable limit. Since SIoV is a cloud based cyber-physical system where all the subsystems are somehow communicating, hence dynamic adaptation of any subsystem may affect the entire system. There can be a range of adaptation strategies to choose from for every SIoV subsystem. Following sections describe some example adaptation strategies for the different SIoV subsystems.

5.3.1 Vehicle Adaptation Strategy

Every vehicle has direct local knowledge of vehicle communication range (r_v), speed (v), and data rates ($[\lambda]$) over the vehicle arrival rate (ρ) and infrastructure inter-distance (L) that are global knowledge. As long as the vehicle workload (ψ_v) is below the threshold (ψ_v^{th}) level, data collection can increase (δ_V^+), otherwise it is curtailed (δ_V^-) (Algorithm 1).

In order to increase data collection (Θ^+), additional safety information can be stored (λ_b). Other option is to increase the virtual communication range (r_v) to allow bigger platoons, so that more interactions can occur among the vehicles. On the other hand, data collection can be curtailed (Θ^-) by reducing additional safety data (λ_b) or by reducing the communication range (r_v). If the infrastructure distance (L) is very long then the vehicles can elect (Θ^*) leader of a platoon who will only handover the bulky message to the infrastructure. Such a setup will allow the vehicles to ignore some overhead interaction messages.

Algorithm 1: Adaptation strategy for the vehicles (V)

Input: r_v, v, $[\lambda]$, `/* local knowledge of vehicle */`
ρ, L, `/* global knowledge for the vehicle */`
ψ_v^{th}, `/* acceptable vehicle workload */`
Output: Workload, ψ_v
begin
    while $\delta_V^{\equiv}(\psi_v, \psi_v^{th})$ **do**
        if $\delta_V^-(\psi_v)$ *needs to curtail* **then**
            if λ_b *can be curtailed* **then**
                $\psi_v \longleftarrow \Theta_V^-(\psi_v, \lambda_b)$
            else if r_v *can curtail* $\vee$ *(v is low $\wedge$ ρ is high)* **then**
                $\psi_v \longleftarrow \Theta_V^-(\psi_v, r_v)$;
            else if L *is too long* **then**
                $\psi_v \longleftarrow \Theta_V^*(\psi_v, leader)$;
        else if $\delta_V^+(\psi_v)$ *can increase* **then**
            if r_v *can be increased* $\vee$ v *is high* $\vee$ ρ *is low* **then**
                $\psi_v \longleftarrow \Theta_V^+(\psi_v, r_v)$;
            else if λ_b *can be increased* **then**
                $\psi_v \longleftarrow \Theta_V^+(\psi_v, \lambda_b)$;

5.3.2 Infrastructure Adaptation Strategy

For the infrastructures r_i, ρ and L are the local knowledge and the objective of adaptation strategy is to balance ($\delta^{=}$) the infrastructure workload (ψ_I). As more sensory information ensure better understanding of the events, hence more data collection (δ^{+}) is preferable. Infrastructure can instruct the vehicles to increase r_v range or to collect more safety information λ_b, which are automatic choice for more (Θ^{+}) data collection. If workload needs to be curtailed (δ^{-}) then traffic can be diverted to alternative routes, which reduces arrival rate (ρ). Also, r_v reduction and leader election can reduce infrastructure workload (Algorithm 2).

Algorithm 2: Adaptation strategy for the infrastructures (I)

Input: r_i, ρ, L, `/* Local knowledge for infrastructure */`
ψ_I^{th}, `/* acceptable infrastructure workload */`
Output: Workload, ψ_I
begin
  while $\delta_I^{=}(\psi_I, \psi_I^{th})$ **do**
    if $\delta_I^{-}(\psi_I)$ *needs to curtail* **then**
      if ρ *is high* **then**
        $\psi_I \longleftarrow \Theta_I^{-}(\psi_I, \rho)$;
        $\psi_I \longleftarrow \Theta_I^{-}(\psi_I, r_v)$;
        $\psi_I \longleftarrow \Theta_I^{*}(\psi_I, leader)$;
      else if L *is too long* **then**
        $\psi_I \longleftarrow \Theta_I^{*}(\psi_I, leader)$;
    else if $\delta_I^{+}(\psi_I)$ *can increase* **then**
      if r_v *can increase* $\vee$ v *is high* $\vee$ ρ *is low* **then**
        $\psi_I \longleftarrow \Theta_I^{+}(\psi_I, r_v)$
      else if λ_b *can increase* **then**
        $\psi_I \longleftarrow \Theta_I^{+}(\psi_I, \lambda_b)$

5.3.3 Home Adaptation Strategy

The number of friends (n_h) one home interacts at any time is related with the possible relationships. If n_h is near the limit then the relationships can be prioritized and higher priority interactions will be given longer slots (Algorithm 3). The home subsystem can handover the sensory information more often and reduce the workload if the threshold level is very close. Another option can be, HBU works as a gateway for personal sensors of the vehicle owner. Personal sensors (e.g., Twitter/Facebook data) can improve the quality of the vehicle sensory data with additional human tagged details.

Algorithm 3: Adaptation strategy for the homes (H)

Input: n_a, n_h, `/* Possible number of friends, Number of Friends */`
ψ_h^{th}, `/* acceptable home workload */`
Output: Workload, ψ_h
begin
    while *true* **do**
        if $n_h >= relation\ limit$ **then**
            $\psi_h \longleftarrow \Theta_h^-(n_h, priority)$
        else if $|\psi_h - \psi_h^{th}| >= limit$ **then**
            $\psi_h \longleftarrow \Theta_h^-(\psi_h, handover)$
        else if *personal sensors available* **then**
            $\psi_h \longleftarrow \Theta_h^*(\psi_h, gateway)$

5.4 Summary

SIoV is a cloud based cyber-physical system to tackle the Internet of vehicles related issues in a more scalable and distributed manner. The cyber layer of the SIoV eases the integration of cross domain CPS services. In this chapter, we have designed the data workload models of the subsystems involved in the SIoV interaction process. The data workload models for vehicle, infrastructure, home, and the cloud would help to understand the storage or computing requirements, which is vital for Big Data management. Cloud computing requirements and costs can be identified by using the above mentioned models. We have provided example adaptation techniques for the SIoV subsystems. The models and adaptation strategies can be useful tools for ITS application deployments.

Chapter 6

C2PS Platform Development and Experimental Analysis

Cyber-Physical System field is in its earlier stage of research and developments. As discussed in the previous sections, contemporary researchers consider IoT as the foundation of the CPS platforms, so that a sustainable future adoption is possible. While CPS is considered as the future of IoT, current technologies and business trends are only at the bay of accepting embedded systems with IP networking (i.e. IoT) capabilities. Real life CPS applications are difficult to develop and test, because of the limitation of deployment infrastructures and their accessibility.

In this chapter, we document our approach to develop a C2PS application development platform published partly in [72] and [74]. The key objective of this platform is to provide a rapid application development and testing procedure in the laboratory setup, so that most of the application source codes can be reused for real life application deployment. In Section 6.1, we provide C2PS platform development details, which is followed by VCPS related developments and SIoV characteristics analysis in the Section 6.2. We also present our approach of *tNote* message development along with some experimental analysis in Section 6.3. In the Section 6.4, we describe a telematics based driving assistance application that follows the C2PS design philosophy. We also present a relevant multi-sensory dataset organization approach, and some applications based on the dataset and the custom built platform.

6.1 C2PS Platform Development

We have developed the C2PS platform following client[1]-server[2] communication model. We have chosen JAVA to develop the platform core so that the implementation can be easily deployed to multiple platforms. The core of every C2PS setup is a *Command Center* that manages the operations in the system. We provide a *SuperCommandCenter* model that can be easily extended to incorporate domain specific properties. For example, in Figure 6.2 we have extended it for VCPS setup using *IovCommandCenter*. All the smart things involved in a CPS operation are extended from *MyElement* class. This class has necessary basic properties for ID management, data storage management and network communication features builtin. All the P2P smart objects in our platform extend this class to get all these features automatically.

6.1.1 Package Structure

Whenever a new smart object joins a C2PS setup, it first informs the responsible *CommandCenter* about its registration. Based on the C2PS domain and the type of the smart object command center specific handlers manage the new smart thing. A command center only sends configuration commands to the participating smart objects such as create connection with another object, disconnect from the session, quit entirely, etc. Any object to object connection is directly P2P between two smart objects either through the physical or the cyber layer. For a cyber layer, each smart physical object is aware about its twin virtual object. In Figure 6.2, for a SIoV setup, there is a *VirtualVehicle* for a physical

[1]MCRLab CPS Client - https://bitbucket.org/kmalam/iot_android
[2]MCRLab CPS Server - https://bitbucket.org/kmalam/iot_server

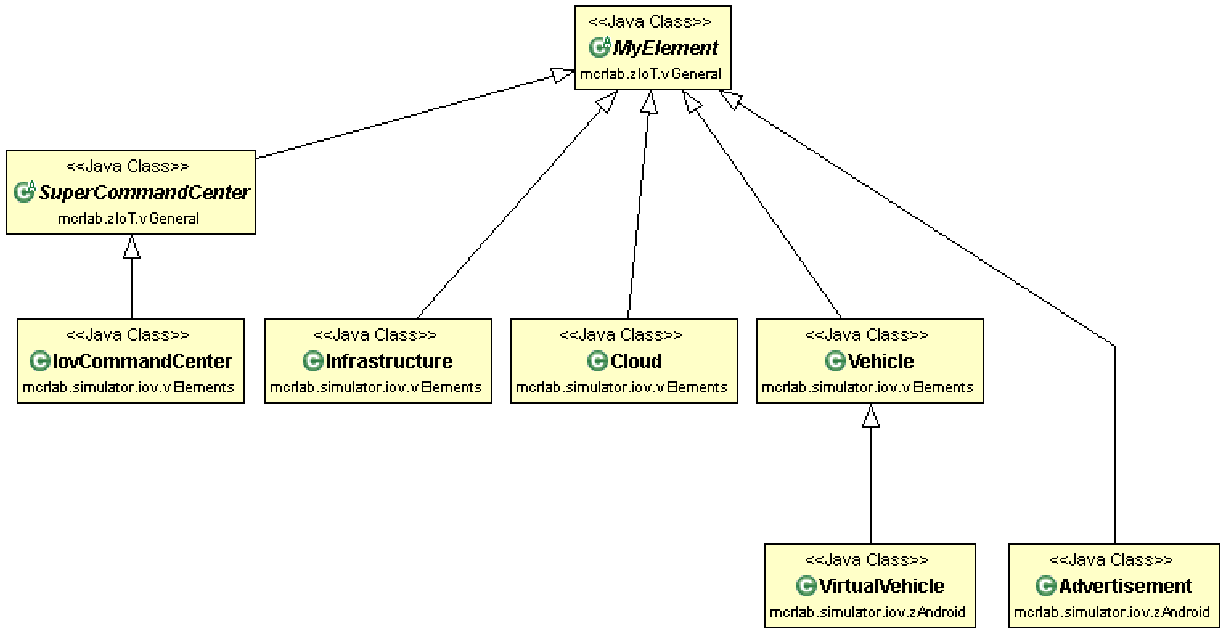


Figure 6.1: The key systems elements of the C2PS platform

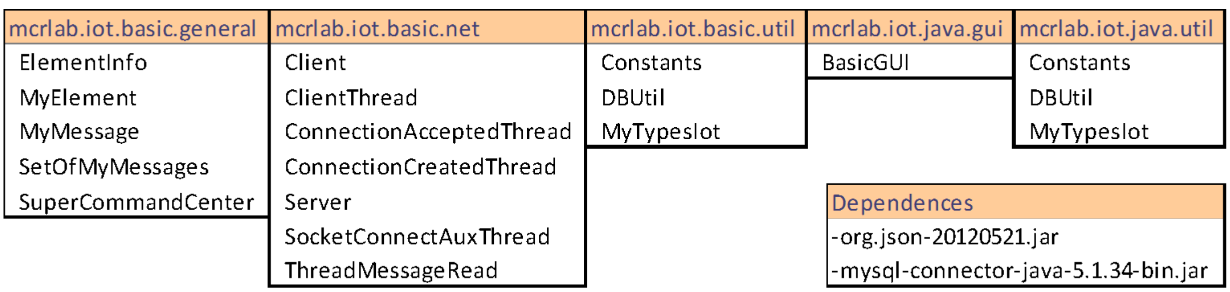


Figure 6.2: The key packages of the C2PS platform implementation.

Vehicle. In this case, the *Vehicle* object is hosted in a mobile application (e.g. Android Smartphone) and the *VirtualVehicle* is hosted as a P2P process in the cloud infrastructure.

The key library `mcrlab.iot.basic` is composed of three JAVA packages that are further extended to support domain specific requirements.

- `mcrlab.iot.basic.general` has the basic elements and message classes.
- `mcrlab.iot.basic.net` is responsible for network connection, messaging, and thread handling.
- `mcrlab.iot.basic.util` provides some utility functions such as database manage-

ment features.

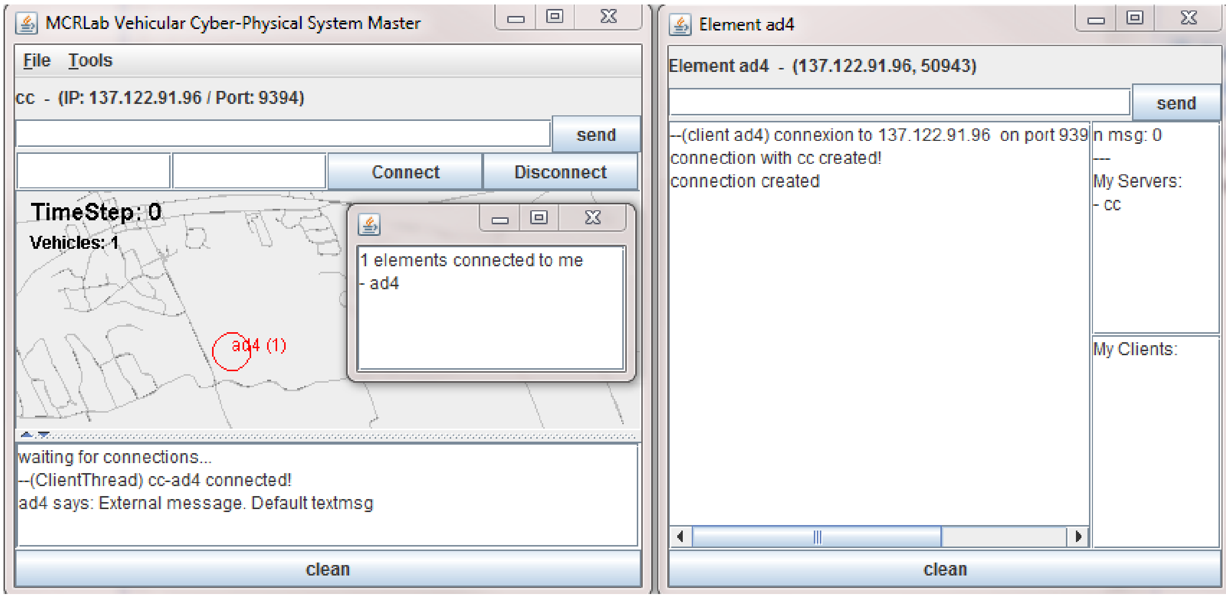


Figure 6.3: Basic GUI layout for a smart object.

There are some dependencies in the platform. We use JSON library (`org.json.jar`) for message exchanges among the processes and MySQL library (`mysql-connector-java.jar`) for database related operations on the MySQL databases available for each smart thing. In the platform, for every P2P system, `Server` class is always waiting to receive connection requests from the clients. For every connection, a new `ClientThread` is created to manage the connection. Every `ClientThread` responsible for a network connection is always listening for a message exchange operation. The message exchanged between every client, and server is formatted following JSON rules. Every new message is handled by a new thread, otherwise message handling fails. To handle these heavy thread operations, it is recommended to have a cloud setup where virtual machines of higher resources can be easily created. Every smart thing comes with a `BasicGUI` (Figure 6.3) that helps to manage different types of connections.

6.1.2 Features

The platform can establish direct connections among the smart objects when physical communication is available, and through a public cloud when not available. When they cannot connect directly, each physical *thing* connects to the intermediate cyber process that is hosted on the cloud and P2P inter process communication is established analogous to physical ad-hoc networks. Following are the available features of the platform at present.

- The platform helps two smart objects to communicate with ease. To do that, the smart objects connect through sockets using a standard client-server connection. Once the objects are created, the platform handles all the connection with a simple call: `smartObject1.connectWith(ip2, port2)`
- At present, we use TCP (Transmission Control Protocol) for the communications but the future version will introduce UDP (User Datagram Protocol) as well to support the low end devices.
- To create a smart object, it is only necessary to extend `mcrlab.iot.MyElement` class, which comes with default communication capabilities and then communication is done as `smartObject1.sendTo(smartObject2, Message)`
- The `Message` in the platform is presented in a JSON format which helps in data sharing with other platforms.
- The platform can handle communications among multiple devices at the same time. It can broadcast or multicast messages to a group of objects or can forward to a specific target object.
- Actuation decisions based on the exchanged messages can be taken directly in the physical things or in the cyber layer by the virtual processes which will be forwarded to the physical things.

- Every smart object comes with an optional MySQL database to store all the exchanged messages along with the metadata.
- One smart object can have multiple relationships with different objects at any time. This is implemented using rules in the connection process. Every connection is tagged by an id such as `Id1_Id2_Proximity_1`, `Id1_Id2_Local_Friend_1`, `Id1_Id2_Remote_Friend_1` etc.
- Currently active relationships (e.g. `Proximity`) and possible participating objects are identified and contacted by the *Command Center*.
- Following message exchanges are supported in the platform: to a targeted object, to all the connected devices, by types of objects (i.e. all the vehicles, infrastructures, cloud, etc.), by the relationship of the objects.

6.2 VCPS Related Developments

In order to prove the usefulness of the platform, we have developed the SIoV emulation setup using the platform. In our setup, we have considered vehicles, infrastructures and cloud storage as our physical *things*. In this case, vehicles are mobile things and both infrastructure, cloud are static things. Mobility traces of the vehicles are generated using SUMO [98] and real life map information collected from the OpenStreetMap[3] is fed to the SUMO to generate vehicles in that map. Random runs of the SUMO simulation generate different vehicle mobility scenarios on the same map data. Infrastructure and cloud positions in the map and their inter connections are augmented using separate XML files from the SUMO mobility XMLs (Figure 6.4).

[3]https://www.openstreetmap.org

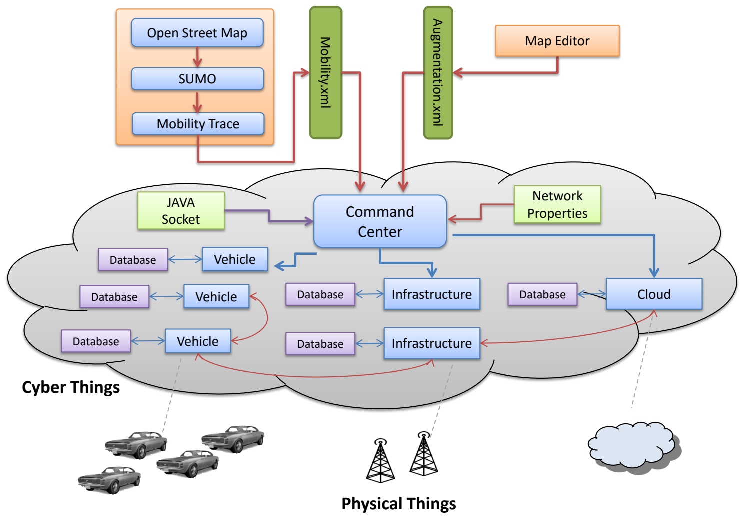


Figure 6.4: SIoV emulator setup, where OpenStreetMap, SUMO mobility traces are input to the C2PS platform and C2PS manages virtual process generations, JAVA IP networking based inter-process communications.

6.2.1 SIoV Emulation Setup

As noted earlier, all the physical things first register with the *Command Center* to which they are assigned with. We can change the settings of the platform using `Properties` file. Once a registration is successful from a physical *thing*, a representative virtual process is created for that physical *thing* in the server cloud by the *Command Center* (Figure 6.5). Then the corresponding virtual process and the native process of the physical thing start a one-to-one communication channel. *Command Center* follows SUMO simulation scenaio time step one at a time and checks the current positions of the vehicles, infrastructures to find possible V2V and V2I interactions. Every vehicle or infrastructure is assigned with default communication range using the settings file that can be updated independently. All the infrastructure processes are always connected with the cloud process to handover data after a settings file based time period.

When it is possible to create V2V connections between two processes, *Command Center* sends a command to one of the vehicles processes to become server and asks the other one to create P2P connections between them as a client. In this process, a bigger vehicle platoon network can be established with no topology limitation. As a result, both single hop and multi-hop communication become possible among a virtual process platoon. The platform can be configured for different type of messaging (e.g. broadcast, single hop).

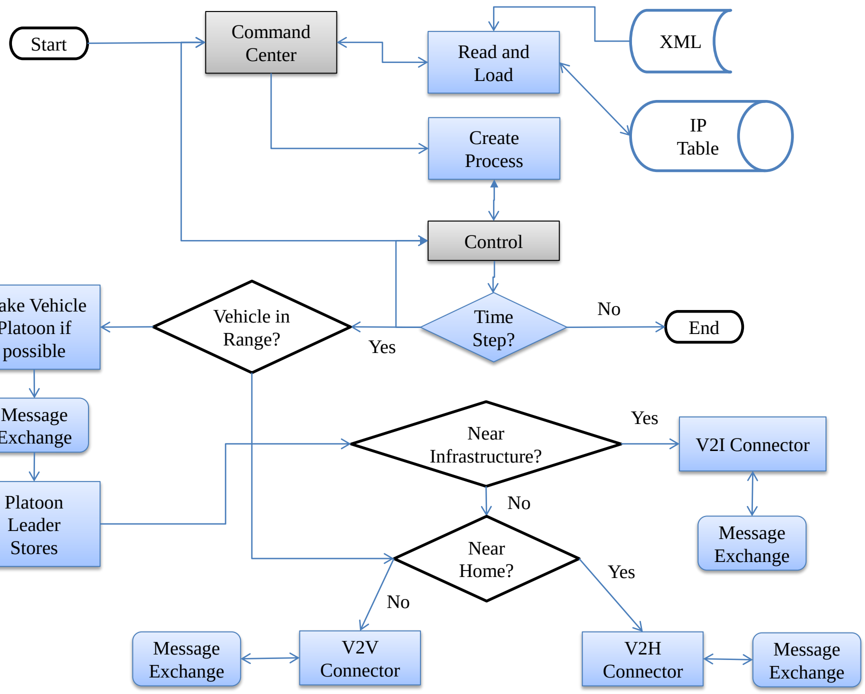


Figure 6.5: A complete SIoV emulation life cycle using the platform.

Every vehicular message can contain the current geo-location, speed, direction, or random vehicular situation from generic OBD II DTC codes. V2V interactions are finally stored in the corresponding MySQL database of each vehicle process. Generally, data is first stored in a cache, and later transferred to the MySQL database. Frequency of vehicle messages can be controlled from the settings file along with the game time step. A useful game time step (e.g. 1 sec $\sim$ 5 game sec) is selected so that heavy thread and network communication can workout through in the platform. Figure 6.6 shows an example setup of the SIoV emulation scenario.

Whenever, a vehicle comes in the range of an infrastructure, *Command Center* initiates a V2I interaction command. As a result, an infrastructure process establishes P2P connection with a vehicle process. At this moment, one vehicle starts to transfer its cached messages to the infrastructure. Once the handover is complete, vehicle sends the cache content to its own MySQL database and clears the cache to receive the next batch of V2V interactions. An infrastructure process receives many such V2I data handovers and stores them initially in a large cache. After a predefined time period is over for an infrastructure, it handovers the cache data to a cloud process and stores data to its MySQL database. At this moment, an infrastructure process is ready to receive the next batch of V2I interactions. So, it clears its current cache. On the other hand, cloud process also maintains a big cache and periodically sends some data from the cache to its MySQL database.

In this procedure, our platform stores data in vehicle, infrastructure and cloud level processes both in primary and secondary data stores. As a result, an entire SIoV emulation scenario is established using the developed C2PS platform. We can develop a different set of safety, non-safety VCPS applications on top of these primary and secondary data storages. We keep the home part of the SIoV implementation for future works, because it requires a different type of simulation from the SUMO. However, implementations similar to the infrastructure can resolve this issue. We can also emulate near real life SIoV deployment, by distributing some of the vehicles, infrastructures, cloud processes in different computers in a laboratory network to reduce cloud hosting costs.

6.2.2 SIoV Characteristics Analysis

In this section, we study the system workload behaviour using simulation and modeling results for the different parameter settings of the SIoV system. Based on the study, we identify the relationships among the system parameters and the subsystem workloads, which are used to design adaptation techniques for the SIoV subsystems.

Simulation Settings

For the simulation measurement, medium scale vehicle mobility traces are generated using SUMO, which are further customized by changing the simulator parameters such as vehicle arrival rate (ρ), and vehicle speed (v). Later, socket based network communications among the subsystem processes are managed by a central process by using the vehicle communication range (r_v), inter infrastructure distance (L), and the RSU communication range (r_i) parameters. We follow the cyber layer based communication model for this simulation.

In our simulation setup (Figure 6.6), every physical element (*i.e.*, vehicle, infrastructure) is represented using a virtual process that can engage in peer-to-peer inter-process communication while running in the cloud. The network communication module of the simulator uses the geographical information from the mobility traces and applies communication parameters so that the virtual elements can establish ad-hoc or static network connections and exchange JSON (Javascript Object Notation) data among them. We consider the network connections to be error-free and data generation rate (λ) of 512-to-1024 bytes/s. All the generated and shared data are stored in the respective MySQL database of every virtual element. Once a virtual vehicle reaches the communication range of a virtual infrastructure, it establishes a network connection between the virtual processes and later hands over the data to the infrastructure. In our simulation setup, arrival rate (ρ) varies from 0.4 to 2.0, vehicle speed varies from 60Km/h to 140Km/h, vehicle communication range (r_v) varies from 50m to 350m, infrastructure density varies from 500m to 5000m. For each simulation scenario, we collect data for 100 executions and report their average.

Relationship of Subsystem Storage Requirements

From the Figure 6.7 we see the data relationship of different subsystems. Since a vehicle is the only data generator hence the storage requirements of the infrastructure, and the home are proportional to the vehicle (*i.e.*, $\psi_I \propto \psi_v$ and $\psi_h \propto \psi_v$). For this test, we consider $v = 16.67\ m/s$, $L = 1000\ m$, $r_v = 200\ m$, $r_i = 500\ m$, $n_h = 20$, and arrival rate ρ is varied. Vehicle arrival rate (ρ) 0.5 denotes that one vehicle is arriving to the scenario in 2s. With the mentioned setup, the platoon size grows with the increment of arrival rate. For example, at $\rho = 0.4$ platoon size is 5 and at $\rho = 2$ platoon size is 24. Platoon size (P)

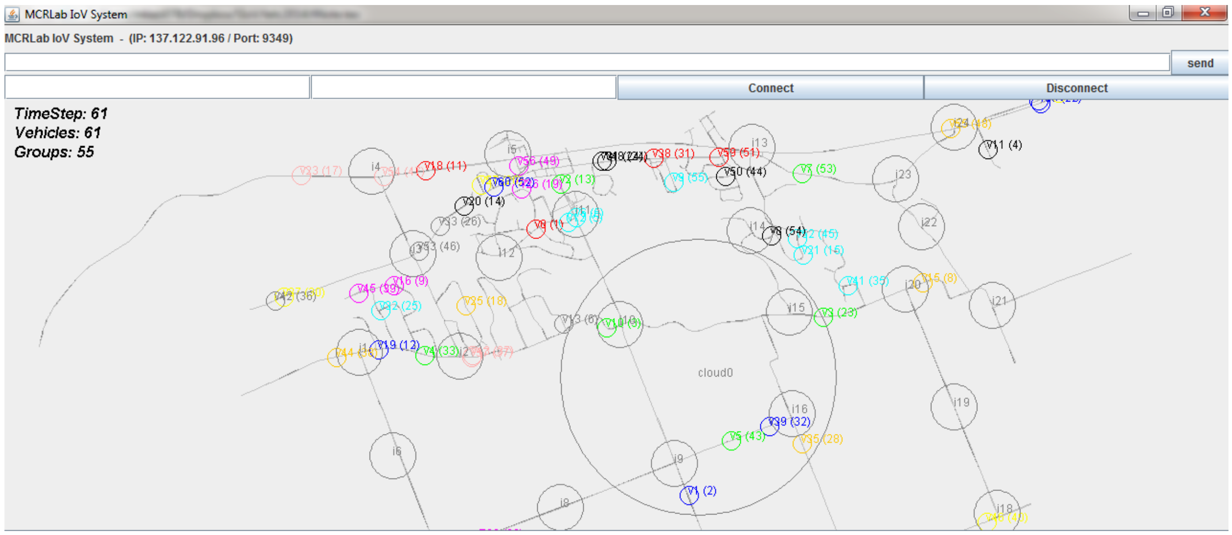


Figure 6.6: In this simulation window, we present 100 simulated vehicles, 25 infrastructures and 1 social graph cloud on the *Cumberland, Ottawa, Canada* region. In this picture, *i(x)* represents infrastructure process, *v(x)* represents vehicle process, cloud0 represents social cloud process, and same colored *v(x)* vehicles are connected in a platoon as *v(x)(y)*.

is the number of vehicles that are connected to exchange information at any time.

Again, from the Figure 6.8 we see that with the increase of data rate the storage requirements grow and with the decrease of the data rate, storage requirements diminish ($\psi \propto \lambda$). For example, if we decide to store all the generated safety messages, then the storage requirements grow, but if we only store the safety messages when there is any important event, then the storage requirements decrease. As a result, we can control the workload of the overall system by storing the valuable information only.

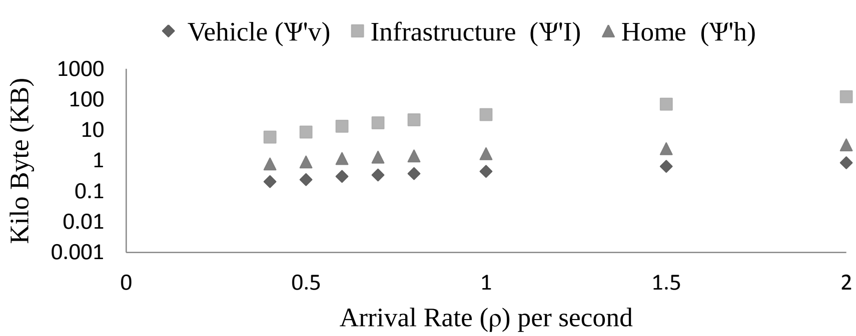


Figure 6.7: The storage requirements of vehicle, infrastructure and home at various arrival rate (ρ) based on the model. Y-axis is measured in log scale of base 10 (thousands unit).

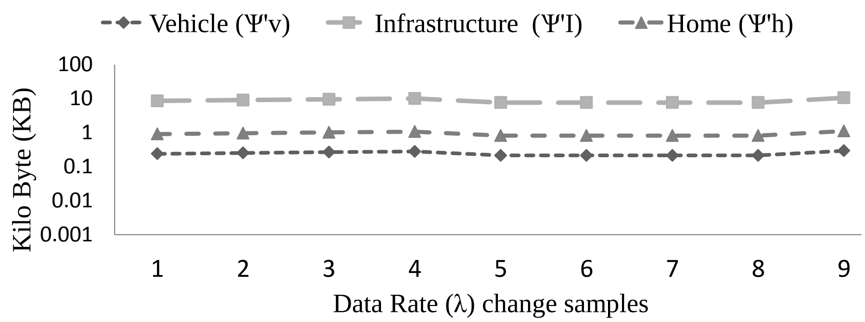


Figure 6.8: Comparing the storage requirements of vehicle, infrastructure and home on the change of data rate (λ). From sample 5 to 8, safety information is not stored. Y-axis is measured in log scale of base 10 (thousands unit).

Effect of Arrival Rate (ρ)

In the Figure 6.9, we compare the storage requirements of vehicles based on simulation results and the modeling results. In this case, we consider $v = 16.67\ m/s$, $L = 1000\ m$, $r_v = 200\ m$, $r_i = 500\ m$, $n_a = 20$ but ρ is varied. The simulation results are very close to the modeling results because both the model and simulation considers average number of platoon members. With the increase of arrival rate the storage requirement also grows ($\psi \propto \rho$).

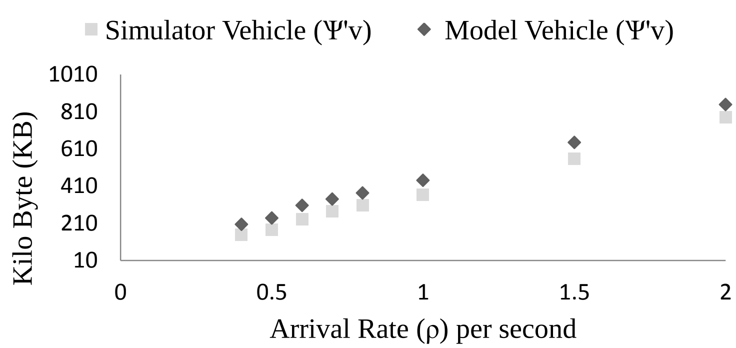


Figure 6.9: Comparison of the simulation and the model based vehicle storage requirements at different arrival rate (ρ).

Effect of Vehicle Speed (v)

In case of vehicle speed effect, we keep arrival rate to $\rho = 0.5$, vary the vehicle speed from 60 $km/h = 16.67\ ms^{-1}$ to 140 $km/h = 38.89\ ms^{-1}$ and keep intact the rest of the parameters as described in the earlier section. We observe from the Figure 6.10 that the storage requirements decrease with the increment of speed ($\psi \propto \frac{1}{v}$). It is to be noted that with higher v measurement time T reduces. From the figure, we see that the model results are very close to the simulation results. With the increment of speed, the connection life of vehicles reduce. As a result, the platoon members interacting at any time falls which reduces the interaction data. In a scenario where vehicles are slowing down due to slow traffic onwards while keeping the same arrival rate and vehicle communication range, the storage requirements surge. When, the speed is almost zero (0.25 m/s) the maximum platoon size grows to around 400.

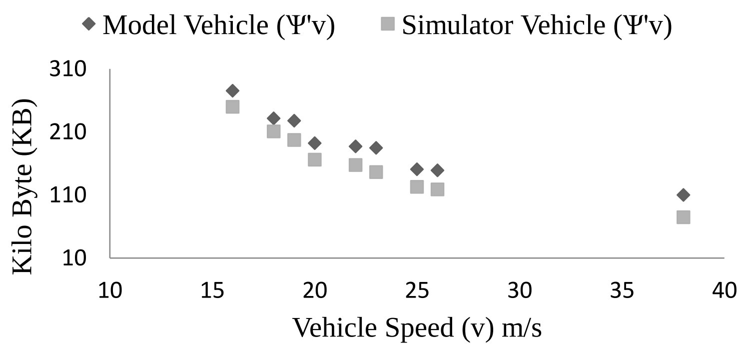


Figure 6.10: Comparison of the simulation and model measurements of vehicle storage requirements in case of varying vehicle speed (v).

Effect of Communication Range

From the Figure 6.11, we see that with the increase of communication range (r_v) of the vehicles the storage requirements grow ($\psi \propto r_v$). This is because, more vehicles join the platoon and exchange information. As a result, each vehicle listens to more of their neighbours. In the given setup ($\rho = 0.5, v = 16.67\ m \cdot s^{-1}$), where r_v is changing from 50 to 350 m, platoon size grows from 2 to 11. If we also vary the infrastructure communication range (r_i) then for higher communication range, more vehicles will be able to report to the infrastructure at any time. The V2I connection time would be longer in this case. The modeling result is very close to the simulation result as well.

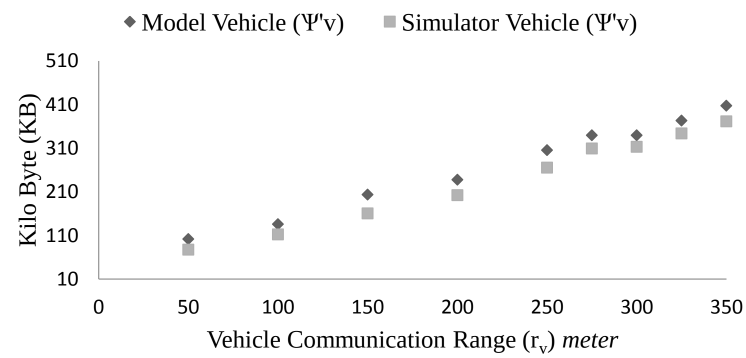


Figure 6.11: Effect of vehicle communication range (r_v) on the vehicle storage requirements in case of simulation and the model.

Effect of Infrastructure Density (L)

When we vary the distance of two consecutive infrastructures, the storage requirements change. From the Figure 6.12 we see, when the distance grows the storage requirements also grow ($\psi \propto L$). As the arrival rate ($\rho = 0.5$) is unchanged and the vehicle speed is also same $v = 16.67\ m/s$, so each vehicle at a time interacts with 5 other neighbours but they travel for longer time to report to the upcoming infrastructure. As a result, ψ_v grows over time.

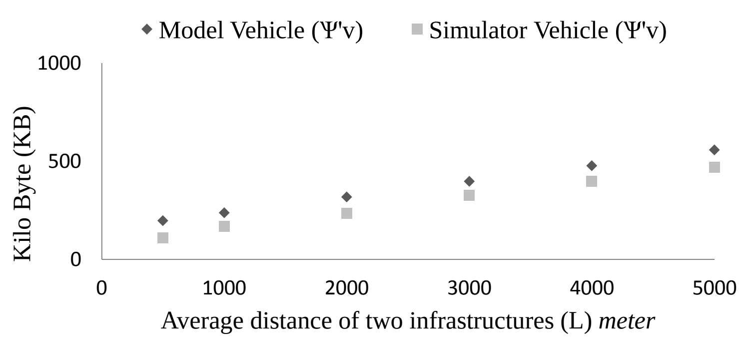


Figure 6.12: Effect of roadside infrastructure density (L) on the storage requirements for the simulation and the model.

Dynamic Adaptation Scheme

In Figure 6.13 we show the experimental results of dynamic adaptation schemes. Since data is generated and transmitted through the vehicles, hence we focus our experiments on vehicles' data workload based adaptation only. We have implemented the algorithm and can select a threshold (ψ_v^{th}) data workload level for the adaptation scheme. Following the threshold level, the algorithm varies different parameters such as vehicle communication range (r_v), basic data rate (λ_b), RSU inter-distance (L) and vehicle arrival rate (ρ). From the figure, we see that the algorithm targets to be around the threshold data level. At any time, we can also update the threshold level that will again change the behavior of the adaptation scheme.

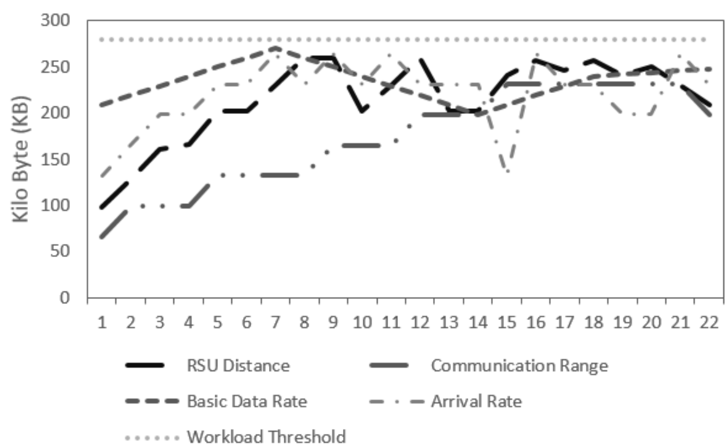


Figure 6.13: A data workload experiment of the vehicular storage for different SIoV adaptation schemes.

6.2.3 SIoV Deployment Setup

Our developed platform can also handle real life VCPS deployment scenarios. In this case, we consider a real vehicle is represented by a Smartphone or Tablet present in that vehicle. It is basically a hub of all the sensors (e.g. OBD Reader, Smartphone, Tablet, Vehicle internal-external sensors, etc.) present in a vehicle. An Android based mobile application is installed in the Smartphone that acts as the native vehicle process. Now, the native process of the vehicle can communicate directly with another vehicle/infrastructure (`physical<->physical`) using `WiFi-Direct` type communication and exchange sensory readings or events. This communication is possible as long as the connection is alive. An example V2V communication setup is the physical layer of Figure 6.14.

The other option is to connect the Smartphone or Tablets to a `WiFi` connection provider (e.g. Router) and communicate through the cloud based cyber layer. In this case, the cloud is considered as a desktop computer connected to the router. The *Command Center* operates in the cloud computer and whenever a new Smartphone client application registers to the *Command Center*, a representative cyber process is created in the cloud. Mobile devices representing the vehicle communication scenario such as Figure 6.14 get to communication with each other using the `physical<->cyber<-><cyber<->physical` route by P2P cyber process communications.

Mobile devices collect values from its builtin sensors and send them to their cyber representative in the cloud. *Command Center* is continuously monitoring the position of the registered mobile devices and help them to establish P2P virtual connections. In this case, `Proximity` relationship comes into play to establish connections. In our platform, we can assign networking properties such as communication range (e.g. 20 meter) to the individual objects (e.g. vehicle, infrastructure). Communication range is associated with the `Proximity` relationship that triggers a data sharing event among the participating objects. Message is organized as JSON file and all the messages are stored in the corresponding MySQL database of the virtual vehicle in the cloud computer. Moreover, random sensory data can be generated from OBD2 DTC information (e.g. vehicle model information, a roadway situation such as accident, road block, slow vehicle, construction, etc.). Cyber layer based communication ensures that the platform is horizontally scalable and different other relationships are also possible.

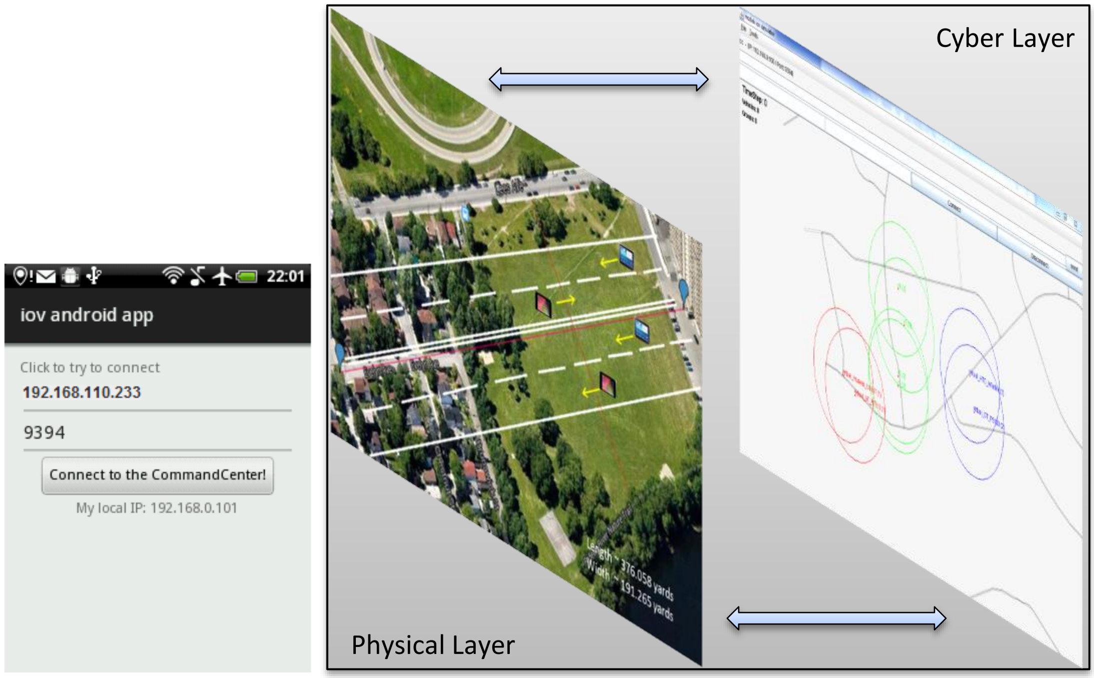


Figure 6.14: SIoV real life deployment setup in a field in the Downtown, Ottawa region.

6.3 tNote Message Development

We have used OSS Nokalva ASN.1/java Studio[4] to develop *tNote* message in ASN.1. We have selected ASN.1 to represent *tNote*, since SAE J2735 message set is defined in ASN.1 and is selected for ITU-T X.680 series of standards [99]. According to ITU, ASN.1 is used in every aspect of our digital life from cellular communication, ATM cash, NetMeeting, RFID, VoIP to biometrics as well as works very well with the XML. We have conducted empirical study to see the impact on *tNote* message payload for various encoding rules. The Basic Encoding Rules (BER), Distinguished Encoding Rules (DER), and Canonical Encoding Rules (CER) type encodings follow *Tag-Length-Value* approach for describing any content where *Tag* represents an ID, *Length* is the length of the message content, and *Value* contains the original message part. For example, in a given code, if `MessageID 12001` corresponds to `80 0A 03 31 32 30 30 31 2E 45 2B 30` hexadecimal bytes, then *Tag*: `80`, *Length*: `0A`, and *Value*: `03 31 32 30 30 31 2E 45 2B 30`. In our implementation of OBU and RSU

[4]http://www.oss.com/asn1/products/asn1-java/asn1-java.html

messages, `RsuTnote` contains a list of `ObuTnote` messages representing OBU-RSU social graph relations. OBU message graph representing OBU-OBU relationship is implemented using `ASN.1 recursion`, i.e. `ObuTnote` can be composed of a list of references to other `ObuTnote`s.

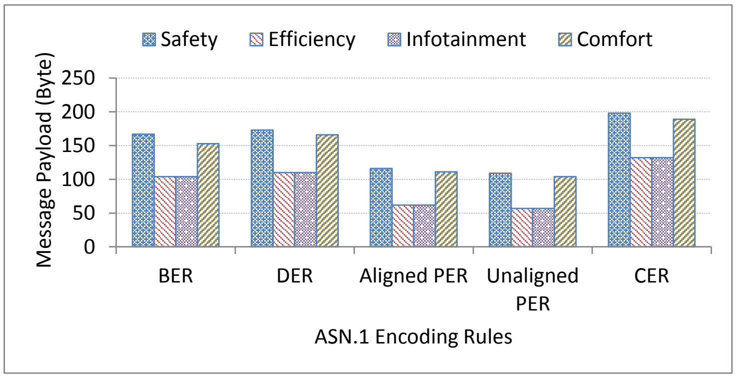


Figure 6.15: Payload size of various *tNote* messages following ASN.1 Byte encoding rules.

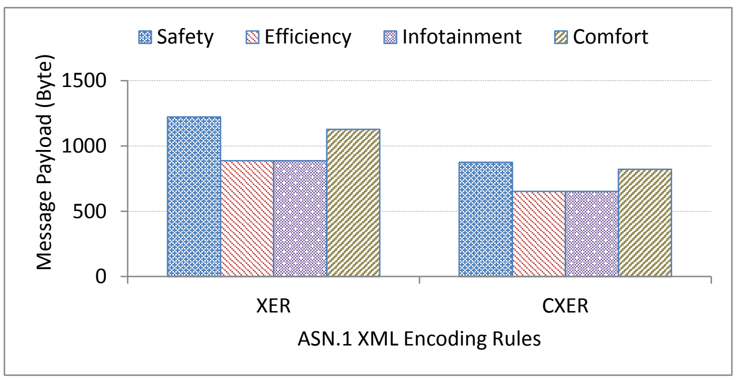


Figure 6.16: Payload size of various *tNote* messages designed following ASN.1 XML encoding rules.

In the Figure 6.15 and 6.16, we find comparative presentation of simpler message payload size for safety, efficiency, infotainment, and comfort applications using variety of X.690 encoding rules for the proposed SIoV. This payload represents the size of one OBU message stored in the OBU storage. The analysis shows that the Packed Encoding Rules (PER) offers the smallest payload whereas BER and DER are larger than PER but almost of similar size. From [99] we come to know that DER[5] type encoding is recommended in the

[5]http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

VANETs communication standards. The XML encoding rules XER and CXER present the same information in a text based format which is easy to work with but very heavy for communication.

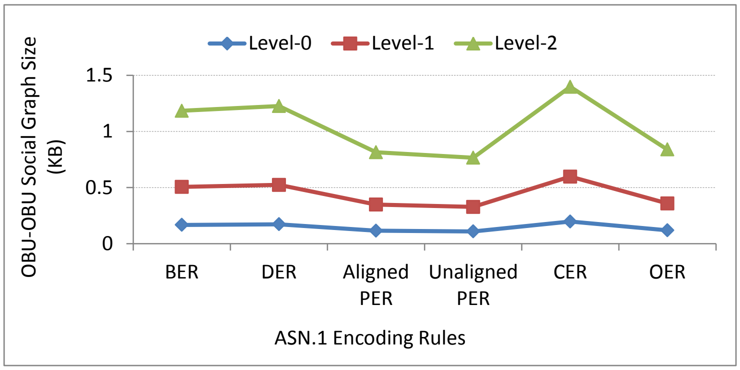


Figure 6.17: ASN.1 data encoding size for OBU-OBU social graph at different friend depths for safety application.

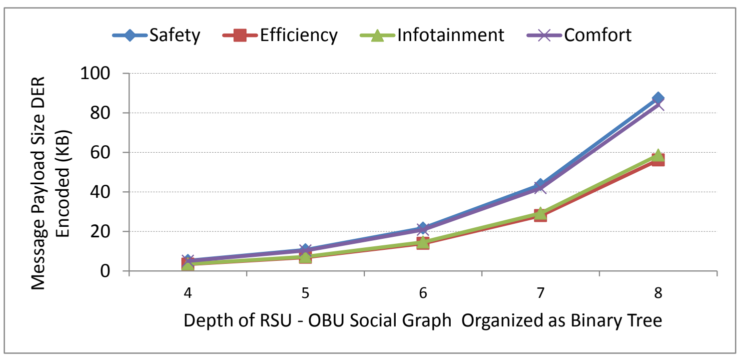


Figure 6.18: ASN.1 data encoding size for RSU-OBU social graph at different friend depths for variety of applications.

The OBU-OBU social graph of Figure 6.17 grows steadily with the depth level increase. For simplicity we consider the social graph to be in binary tree format. For level n, the total number of nodes found in a binary tree is $2^{n+1} - 1$. With level 2, OBU-OBU social graph of a platoon of 7 vehicles stands around 1.2 KB for DER and 0.8 KB for PER type encodings. This OBU-OBU social graph data is transferred from a platoon leader OBU to an RSU. In Figure 6.18, the payload size of OBU-RSU social graph considers tree levels 4 to 8 which corresponds to 31 to 511 vehicles. In between two RSUs there can be multiple platoons. For a platoon size of 511 vehicles, the safety type RSU-OBU social graph can be of 88 * 8 Kb. So considering each service channel is allocated 10 Mbps, one RSU can handle many platoons simultaneously. In the real life scenario, the platoon size depends upon the communication range of the platoon leader and its moving speed [62][100].

In general, a service advertisement message goes through the control channel of VANETs and once a service channel is selected for data communication between the provider and consumer vehicle, then the data exchange starts through the service channel. Hence from the above analysis, we can conclude that OBU-OBU communication in the physical layer can continue to work with the DER encodings. Also, OBU-OBU social graph can be sent from the platoon leader OBU to the RSU using the service channel by following the DER encodings. In case of cyber layer based communication, physical thing can send the DER encoded message to the cyber representative and inter-process communications or updating the receiver physical vehicle can use DER encodings. However, as XML is easy to follow, search and different post-processing tools are easily available, hence OBU, RSU, and HBU can convert their social graphs into XML format before storing them into their secondary data storages. As a result, SIoV system is viable using existing VANETs spectrum and manageable in the cloud.

6.4 Telematics based Driving Assistance Application

In this section, we describe a telematics based driving assistance application for the VCPS (Figure 6.19) that follows the C2PS design philosophy. Our goal is to demonstrate the advantages of C2PS based application designs. In this application, we consider two sources of sensory values: 1) Mobile sensors that capture the user interactions, GPS location of the vehicle, speed, acceleration, etc. 2) On Board Diagnostic II (OBD2) scanner that reads the real time status of the vehicle such as fuel consumption, airbag status, etc. We show the usage of different types of C2PS computations that applies sensors and services fusion to identify various driving events and/or driving related situational recommendations for

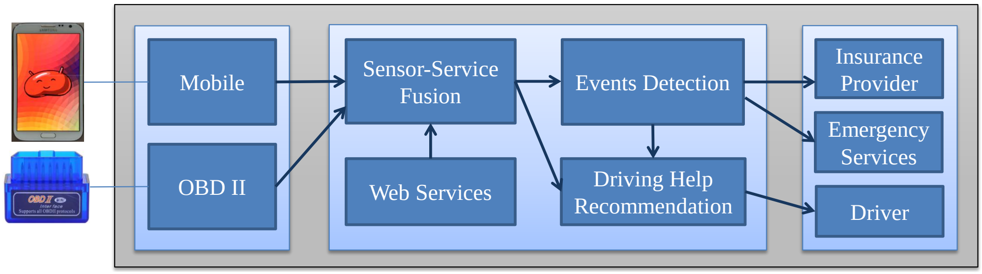


Figure 6.19: Telematics based driving related recommendation system.

drivers, insurance or emergency service providers.

6.4.1 Computation

We have designed the finite state machine (Figure 6.20) of the telematics system using Qfsm[6] that can translate the graphical design to SMC[7] enabled format. SMC file format is later transferred to source code such as JAVA. At first, the system stays at *Data_Reading* state, where it reads data from the mobile and/or the OBDII devices. Later based on the current context of the system one of the *Physical_Processing*, the *Cyber_Processing*, or the *Cyber_Physical_Processing* computation model is selected. This decision is taken by the control part of the system. For the *selectPhysical()* action, all the processing occur in the physical layer that is in the vehicle. Several real time driving and usage events are detected in this level of operation and subsequent driving recommendations are made based on the available sensory data.

In case of the *selectCyber()* action, all the processing occurs in the cloud layer. As a result, this type of processing can not provide real time event detections or driving recommendations. But, time delayed operations can benefit from the cloud computing infrastructure for their horizontal scalability. The *selectCyberPhysical()* action expects a hybrid operation where real time processing occurs in the physical layer and resource heavy processing occurs in the cloud layer. Cloud based processing also provides easier,

[6]Qfsm: a tool to design finite state machines, http://qfsm.sourceforge.net/
[7]SMC: State machine compiler, http://smc.sourceforge.net/

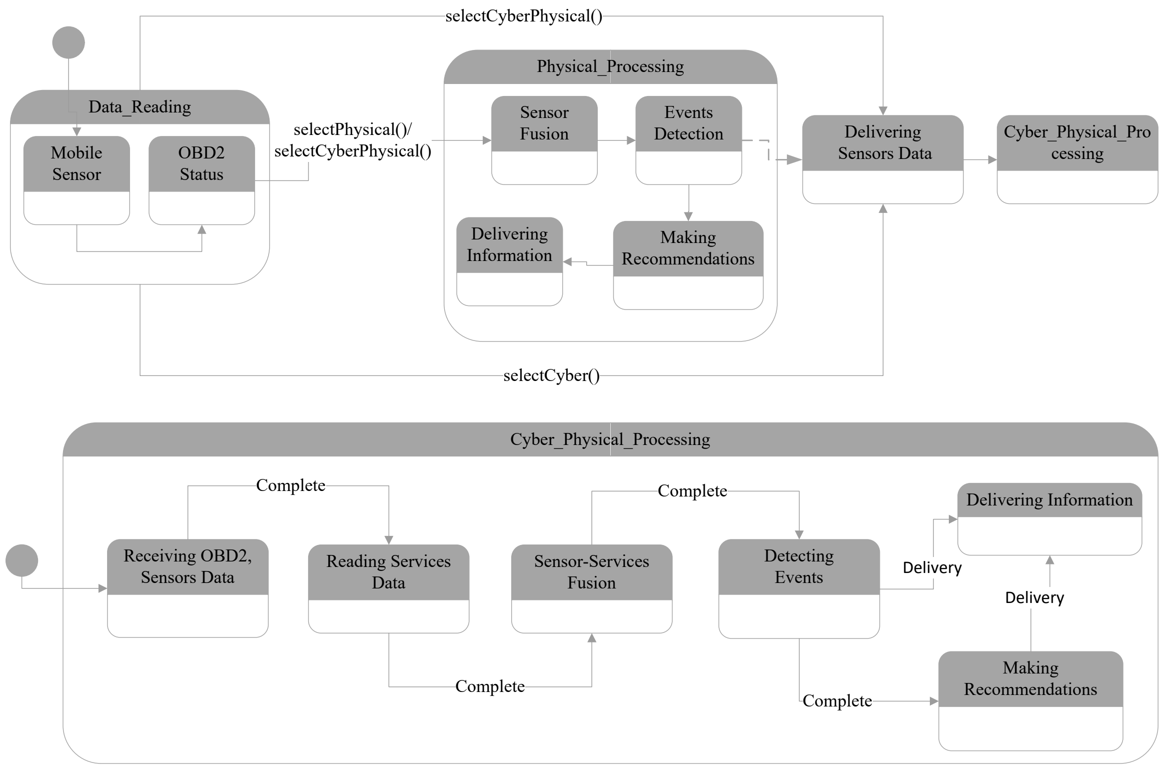


Figure 6.20: A simple state machine representation of the telematics system. (Implemented using Qfsm)

cost effective and timely integration of web services which is not readily available in the physical layer. Both the *cyber* and *cyber-physical* modes enable sensor-services fusion, an extension to the sensor only fusion that is readily available to the physical layer. This enables higher degree of driving support recommendations.

6.4.2 Control

The control part of the system is divided into two sections. The first part is a Bayesian network (Figure 6.21) that takes input about system contexts such as communication range, computation cost, system battery level and communication cost. The Bayesian network decides which one of these two modes (i.e. *cyber-cyber*, *physical-physical*) will be selected. As the *cyber-physical* mode is a hybrid organization, it can have many possible combinations. In order to tackle this issue, we take the previous two opposite modes as

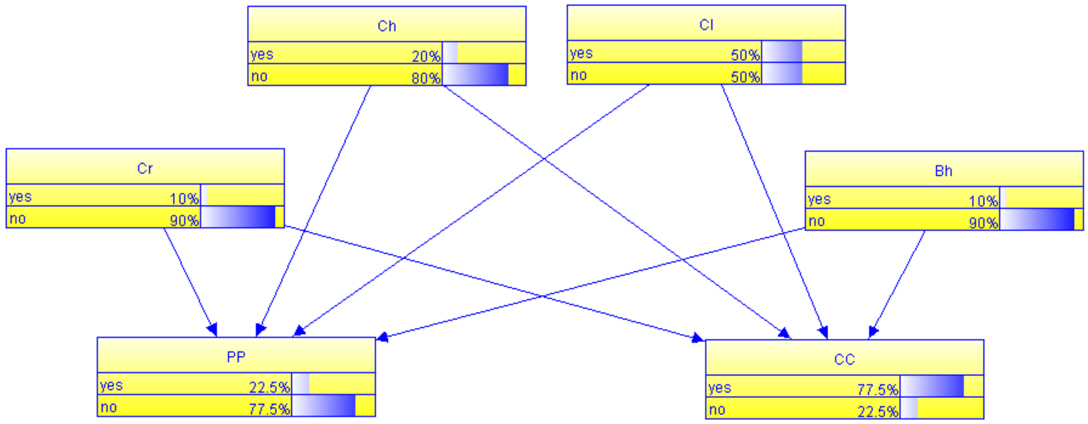


Figure 6.21: Selection of one of the two context based connections using a Bayesian network.

inputs to a fuzzy logic based controller. We have selected fuzzy logic since its rule base can be always updated, which suits the nature of C2PS hybrid computing reconfiguration. Figure 6.22 shows the surface view of the entire fuzzy logic rule base. For this example, we have selected two hybrid computing modes *CP1*, and *CP2*. Some of the rules of this setup are:

R_1: if $P(PP)$ is VH and $P(CC)$ is VL then O^f is ρ_{pp}

R_2: if $P(PP)$ is H and $P(CC)$ is VL then O^f is ρ_{pp}

R_3: if $P(PP)$ is M and $P(CC)$ is VL then O^f is ρ_{pp}

R_8: if $P(PP)$ is L and $P(CC)$ is H then O^f is ρ_{cc}

R_{12}: if $P(PP)$ is L and $P(CC)$ is L then O^f is ρ_{cp1}

R_{15}: if $P(PP)$ is VL and $P(CC)$ is VL then O^f is ρ_{cp2}

Here, *Very High*, *High*, *Medium*, *Low*, and *Very Low* are presented respectively as *VH*, *H*, *M*, *L*, and *VL*.

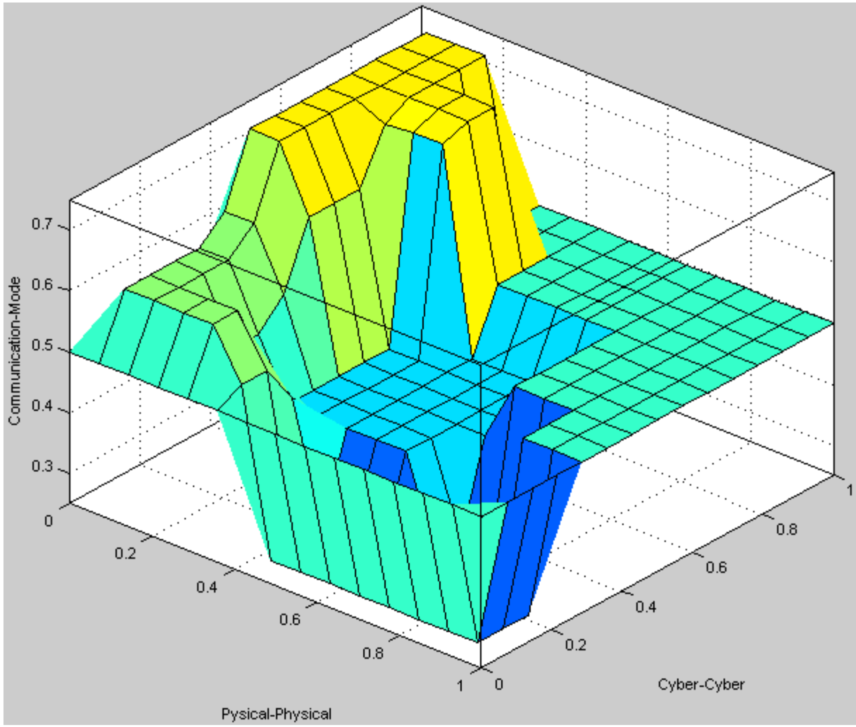


Figure 6.22: Rule base of the fuzzy logic based controller of the telematics based system

6.4.3 Sensors and Services Fusion

Three possible types of data fusions are possible for this situational driving support recommender system that follows C2PS design philosophy. At first, physical layer based sensors only fusion that can provide near real time driving events detection as well as render driving related guidelines to the driver. From Table 6.1 we see that *speeding* event can be detected based on sensory values accelerometer, gyroscope, barometer, GPS or from OBD2 [101]. Similarly, *phone usage* can be identified based on accelerometer, gyroscope, screen time count, text, call metadata. Identification of these critical events can elicit color based warning to the driver through the vehicle dashboard.

Secondly, cloud layer based services only fusion can provide delay tolerant services such as nearby parking or hotel or restaurant information, location based deals, accident statistics etc. [101]. Finally, C2PS hybrid sensor-services fusion can extend both of these sensor or services only fusions. This kind of fusion is a true application of fog computing.

For example, the speeding event detected by the physical layer can further be fused with the location, weather services information coming through the cloud layer and provide location specific speeding related possible fines and/or demerit points. Figure 6.23 shows an Ontario, Canada based model for such an application, where the system can determine the vehicle is in city, village or in other areas and can provide the driver with warnings such as speed based possible fine, demerit points or even accident statistics of upcoming road segment.

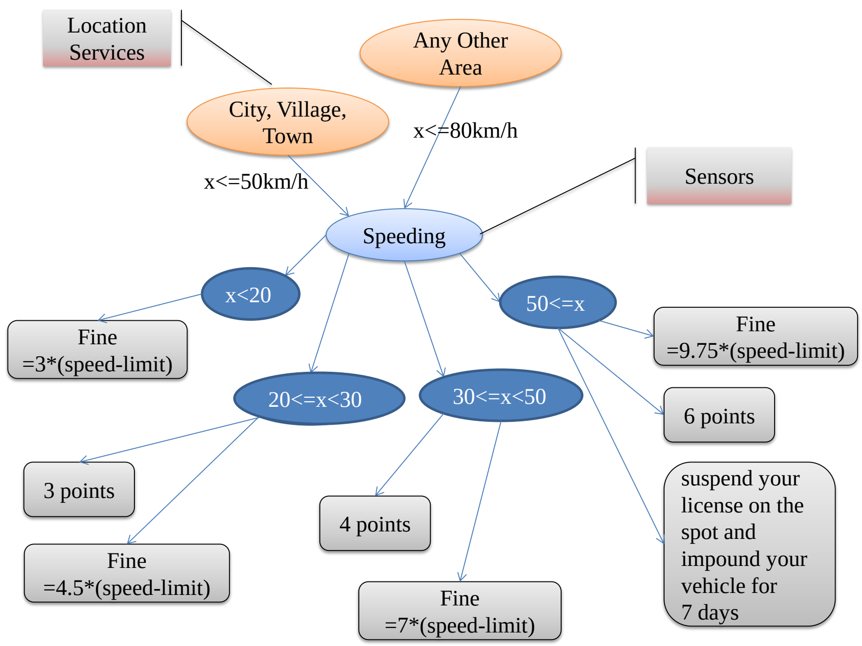


Figure 6.23: An example driving assistance model for Ontario, Canada region based on speeding event.

Information Available	**Sensors**											**Services**
	Mobile										**OBD2**	
	Accelerometer	Gyroscope	Magnetometer	Barometer	Temperature	GPS	Screen Time	Text	Call	Time		
Sharp Acceleration	✓	✓									✓	
Sharp Deceleration	✓	✓									✓	
Phone Usage	✓	✓					✓	✓	✓			
Speeding	✓	✓		✓		✓					✓	
Time										✓	✓	
Turns	✓	✓				✓					✓	
Duration	✓	✓	✓								✓	
Weather Information					✓							✓
Traffic Information												✓
Location Service												✓
Accident Statistics												✓
Airbag Triggered											✓	

Table 6.1: Some vehicular events and their relationships to various sensors and the web services.

Table 6.2: Example vehicular CPS data collection scenarios

Scenario ID	Scenario Objective	Location
Scenario 1	Cooperative vehicular platoon	Ottawa, ON
Scenario 2	Driving acceleration events	Ottawa, ON
Scenario 3	Hill climbing sensory details	Gatineau, Québec
Scenario 4	Downhill sensory details	Gatineau, Québec

6.4.4 Multi-Sensory Dataset

In [102] we describe a VCPS dataset (i.e. MUDVA) architecture that is divided into scenario based modules, where each scenario represents a target objective for the multi-sensory data collection event. Modular architecture enables individual growth of the scenarios. In Table 6.2, we present a small list of vehicular CPS data collection scenarios. Every scenario folder in the dataset describes the structure of that folder using metadata `description.xml` file. It also consists of a `tools` folder that comes with necessary programming functions (e.g. MATLAB). A partially finished dataset is available here[8].

Case Study: Scenario 1 (Cooperative Vehicular Platoon)

In this case study, we have collected data on 29-10-2014 from 16:10 PM to 16:40 PM in the suburbs of Cumberland, Ottawa with varying traffic speeds. The route taken by the cars during the data collection is plotted in Figure 6.24. We used the altitude and longitude values of the first car in the experiment to draw this map. The weather was partly cloudy. The target scenario is a collaborative vehicular platoon of three cars.

Data Collection Setup: Our car platoon had three cars, where each car was equipped with one tablet (Galaxy Tab), one smartphone, and four webcams. All the webcams recorded both video (720p resolution) and audio data. Galaxy Tab and smartphone recorded all the sensors available on the devices including accelerometer, magnetic field, orientation, gyroscope and GPS. All the cars kept close distance with each other while

[8] MUDVA Dataset Link - http://www.site.uottawa.ca/~mkazi078/dataset/

moving. The overall setup and platoon scenarios are shown in Fig. 6.25. The sensor readings were recorded every 0.05 second and saved on the Android devices in CSV format. The video and audio were saved on the laptop in MP4 format.

All the cars started recording the video and sensor readings at the same time. In each car, there were four Logitech HD cameras (720p resolution). The sampling rate of the audio was 8000 samples per second and frame rate of videos was 8 fps. The webcams in the first car were connected to an HP laptop, in the second car were connected to a Toshiba laptop, and the last cars webcams were connected to a DELL laptop. For the first and third car, the Android devices used were the Samsung Galaxy S4 Smartphone and a Galaxy tab GT-P7510. For the second car, we used the same Galaxy tab and Samsung Galaxy Note II Smartphone. To record the video-audio set we used iSpy[9] program while AndroSensor[10] program was used to record the sensor readings.

Figure 6.24: The GPS plot on Google map displaying the route taken by the cars.

Data: The main sensor readings in our case study are accelerometer, magnetic field, gyroscope, orientation and GPS. In Fig. 6.26, we can see a sample of clips from the video recordings, with four clips from each webcam in the three different cars. Below is a brief description of the sensors recorded:

[9]iSpy - https://www.ispyconnect.com/

[10]AndroSensor - http://www.fivasim.com/androsensor.html

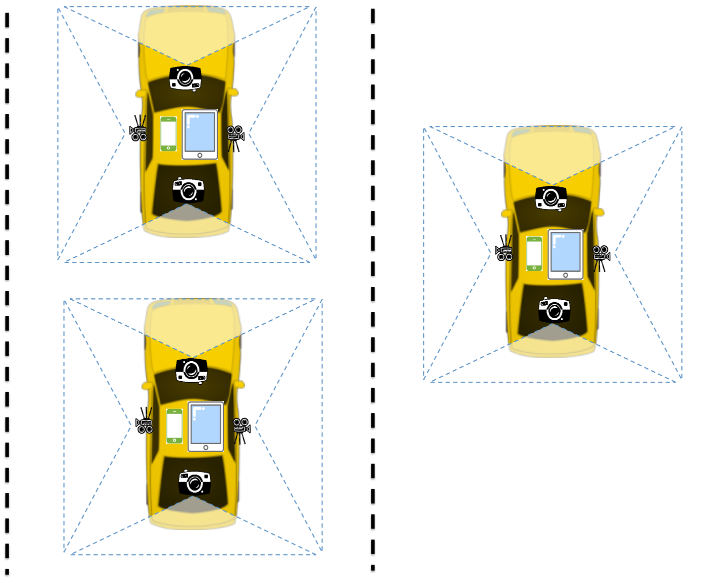


Figure 6.25: Data collection setup of the scenario cooperative vehicular platoon.

- *Accelerometer*: It calculates the acceleration force and gravity force, which are enforced on the mobile devices on all their axes (x, y and z). The unit of measurement is m/s^2. Fig. 6.27 shows the values of the accelerometer on all axes of the devices in the three cars for the duration of 8 seconds.

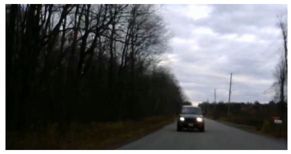
- *Magnetic Field*: Magnetic field is measured in micro-tesla. It gives the direction and strength of the magnetic field relative to the mobile device.

- *Gyroscope*: Gyroscope readings are shown in Fig. 6.29. It records the rotation as rad/second around the three axis x, y and z of the device.

- *Orientation*: The orientation is recorded in terms of azimuth, pitch, and roll. It provides position of a device with respect to the earth's frame of reference (specifically, magnetic north).

In Fig. 6.27, the sensor readings are plotted to show some values for 8 seconds. The starting time of the plot corresponds with the time of the video snapshot in Fig. 6.26. The X, Y, and Z axis follow standard organization from a general android mobile devices.

(a) Front Camera

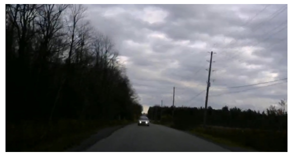


(b) Back Camera

(c) Left Camera

(d) Right Camera

(e) Front Camera



(f) Back Camera

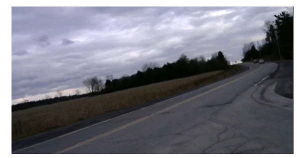
(g) Left Camera

(h) Right Camera

(i) Front Camera

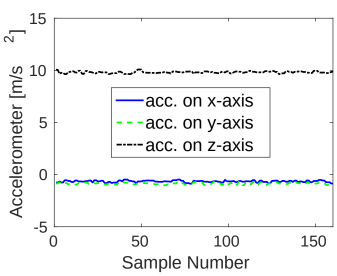
(j) Back Camera

(k) Left Camera

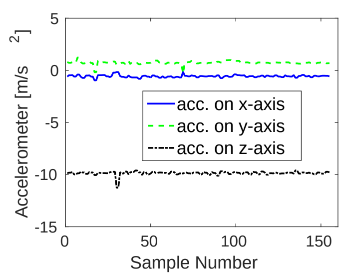


(l) Right Camera

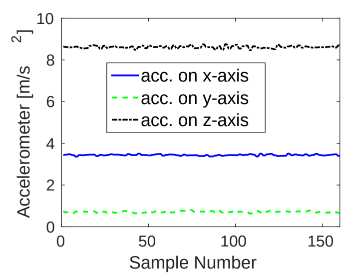
Figure 6.26: Video snapshots from different cameras in cars: a, b, c and d from car 1; e, f, g and h from car 2; i, j, k, and l from car 3.



(a) Car 1

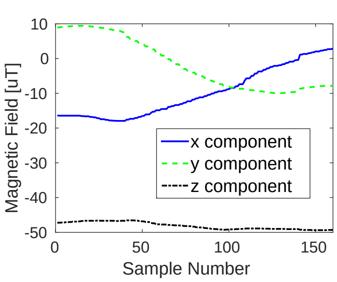


(b) Car 2

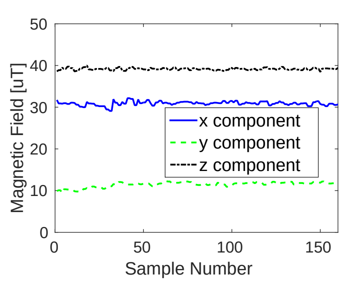


(c) Car 3

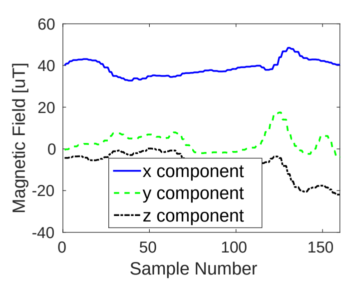
Figure 6.27: Accelerometer readings for each android coordinate axis in the three cars.



(a) Car 1

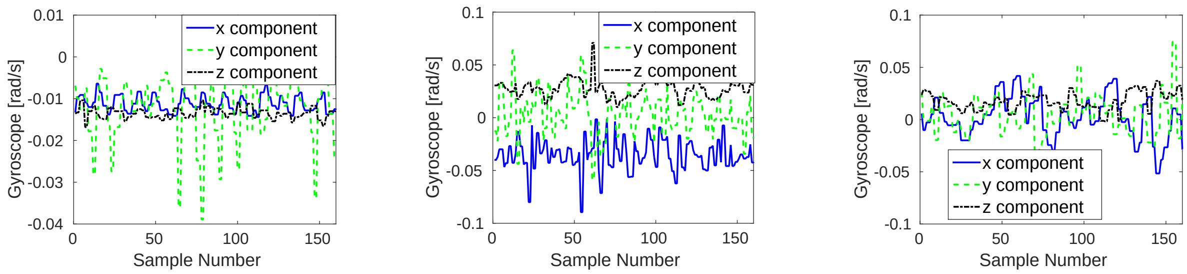


(b) Car 2



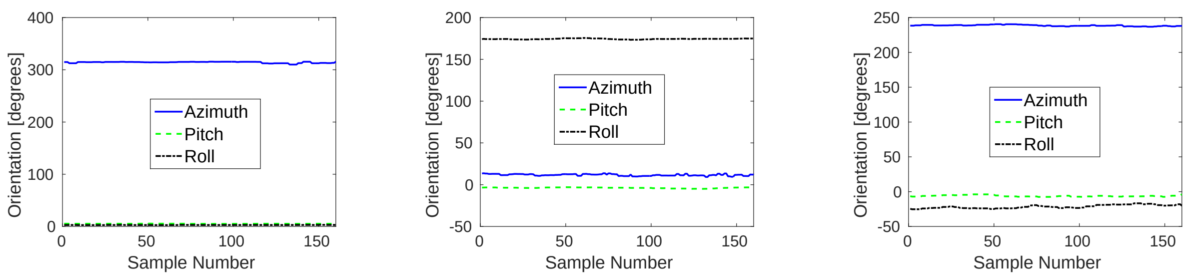
(c) Car 3

Figure 6.28: Magnetic field plot for each android coordinate axis in the three cars.



(a) Car 1 (b) Car 2 (c) Car 3

Figure 6.29: Gyroscope plot for each axis for the three cars.



(a) Car 1 (b) Car 2 (c) Car 3

Figure 6.30: Orientation plot for each axis for three cars.

The tool sets available with the dataset can be used to detect some driving events such as acceleration, deceleration, turn, and no-turn events.

6.4.5 Cooperative Safety Application Based on the Dataset

We use the dataset in our custom built C2PS application development platform ([74]) to emulate message sharing among a platoon of vehicles. We detect acceleration and turn events from the provided dataset, which can be shared with other connected vehicles of a vehicular platoon. The virtual wireless communication range in the platform can be customized (e.g. 200m), which denotes that two vehicles in the close proximity are in the same platoon and will receive each others shared messages (Fig. 6.31).

We use the event detection methodology described in [101]. The power spectrum (PSD) feature of the frequency domain is considered for a hybrid method of supervised (SVM) and unsupervised (Clustering) learning in the process of events detection from the given data. In order to decide on the K-value of clustering algorithms, a combination of heuristic and non-heuristics information about the non-processed data have been considered. Also, it is important to consider the clustering objective and the characteristics of the chosen features of data.

In case of vehicle turn event detection, we were interested in detecting any turns (Turn or Non-turn) from the acceleration values, and did not consider whether one event was a right or a left turn. Therefore, K=2 was selected that resulted in clustering the acceleration data into Turn, and Non-turn clusters. The same method applies for the Acceleration or Non-acceleration events. We used the accelerometer data and selected Acc_y for acceleration events and Acc_x for the turn events detection process.

These events are usually used in the cyber layer of the CPS platform and attached to the *tNote* [70] message along with the sensory values and transmitted to other platoon members that are connected at any time. These sensory values and detected event information is used for safety related decision making in the messaging vehicle or the receiving vehicles.

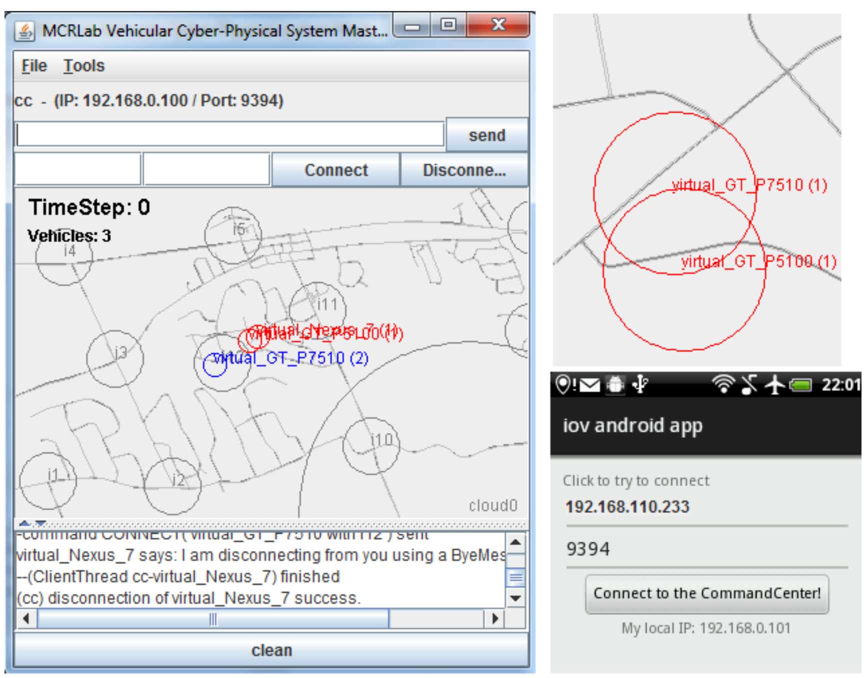


Figure 6.31: Vehicular CPS platform setup for the cooperative safety application based on an occasion of the Scenario 1 dataset.

6.4.6 Location Based Dynamic Advertisements

We have used the same platform to design location based dynamic advertisement application. For example, if a vehicle is passing a restaurant, then the restaurant can draw vehicle passengers' attention by offering special deals in their mobile IoV app (Figure 6.32). In this case, the restaurant has to subscribe for a virtual advertisement node, and the passengers need to have the mobile app along with internet connection.

Every time the IoV advertisement *Command Center* detects the location of a vehicle in the range of an advertisement node, it will ask the vehicle to connect using a command message. Once connection is successful, the advertisement node sends an advertisement to the Android based vehicle app which is displayed one the mobile screen in a dialog. Some of the information of the dialog comes right from the advertisement post within the message.

6.5 Summary

In this chapter, we provide the implementation details of the C2PS application development platform. The platform follows client-server communication model and has been developed using JAVA technology. It follows the inter-process P2P communication model to exchange messages. We use Smartphones and Tablets to represent the physical vehicles, which are loaded with SIoV client applications to connect with the C2PS server. We use the platform to setup usual SIoV communication scenario along with necessary analyses to find the relationships of various system parameters. A small scale real life deployment scenario of SIoV is also discussed. Furthermore, we present ASN.1 based development details for the SIoV interaction message, *tNote*, in addition to related experimental analysis. Additionally, we present the details about a telematics based driving assistance application that follows the C2PS design philosophy, a multi-sensory dataset and a few applications built using the platform.

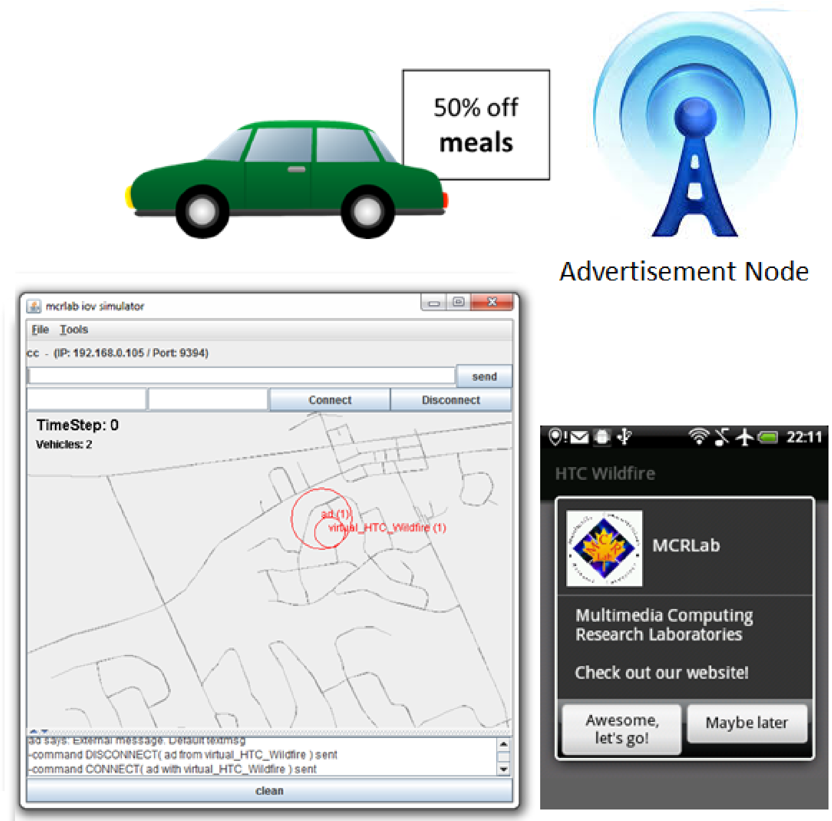


Figure 6.32: Location based advertisement application built using the IoT platform.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

The new trend of ubiquitous devices is ushering a future of connected world. A Gartner report claims that the number of connected devices will be close to 21 billions by 2020. Most of these devices are either used as identifiers, equipped with sensors to collect data about the physical environments or smart enough to receive actuation commands to modify the system or the environmental settings. The availability of smart technology has made the data abundant, which ensures that the gap between data generation and data consumption rapidly diminishes. As a result, this is the golden era of Smart City application deployments. Different types of Smart City applications such as smart transport, smart water/waste/energy/grid management, smart home/office/agriculture, etc. can take the benefits of this data to improve quality of living of the city dwellers.

Cyber-Physical system is considered the next generation of Internet-of-Things, where computation, communication and control features of the physical systems get distributed, and physical devices mostly act as data sources for the computation modules. Computation modules analyse the current context of the system and recommend control actions for the physical environment if required. As a result, there exists a twin feedback loop that is always active to improve the quality of services of the physical systems. This approach further becomes more scalable, once computation and control are featured by the cloud computing infrastructures. In this dissertation, we focus on the vehicular domain of the cloud based cyber-physical systems, VCPS.

In order to describe VCPS, we first introduced a digital twin architecture reference model for the cloud based cyber-physical systems, which is called C2PS. In this case, we divide the system in different operational modes: physical level sensors-fusion mode, cyber level services-fusion mode and various degrees of deep integration of sensor-services fusion modes. We provide a system context based control decision scheme that uses Bayesian network, and fuzzy logic rule base to select any of these system modes for computation and inter-system interactions. We have analytically modeled the computation, communication and control properties of the C2PS. We further follow the model to define the vehicular CPS communication scenarios, where we capture both the intra-VCPS and inter-VCPS communications. A telematics based driving assistance application is also discussed that provides better driving recommendations using the C2PS design philosophy.

While designing the VCPS architecture, SIoV, we consider IoT design principles as a key foundation and also consult the recommendations from social internet-of-things. In this process, we leverage the VANETs technologies and map them with the IoT-Architecture reference model. We have also identified the social structures and interactions possible in case of the VCPS communications. It follows the twin cyber-physical communication model, where it is possible to do communications either through physical-physical communication channel or through the cyber based P2P inter-process communication channel. Additionally, we have presented a wrapper message structure to organize vehicle driver status, vehicular sensory events, and machine-to-machine social interactions among the Internet-of-Vehicles systems.

In every CPS application, independent systems work together to fulfil one common goal. That's why, it is important to enable different levels of reconfigurations in the CPS applications. While selecting from different modes of system operations belongs to self-reconfiguration, one CPS subsystem also needs to have both local, global knowledge about the reconfigurations effects. From this perspective, we adopt model based VCPS reconfiguration approach, where we have developed workload models for the subsystems involved in VCPS interactions. We validate the models using extensive simulations and use the analysis results to find relationships among different system parameters. We use these relationships to provide dynamic adaptation schemes for different VCPS subsystems.

We have developed a C2PS application development platform that follows the client-server communication model. We have used JAVA to develop the server side of the platform, whereas the client side is developed using Android technology. The system follows twin feedback based CPS communication model, where physical devices can either follow physical-physical communication channel or cyber based peer-to-peer inter-process communication channel. We describe the SIoV emulation technique in the lab environment using the platform, since it is difficult and costly to have a real life SIoV setup. We also provide a small scale SIoV deployment technique using the platform. Furthermore, we have developed the wrapper message using ASN.1 and provide related experimental analyses.

7.2 Future Works

The cloud based CPS architecture reference model, C2PS, is a generic model that can be applied to different CPS application domains. In this dissertation, we show how the C2PS model can be fit to the smart transport arena i.e. vehicular CPS. The knowledge gathered from the application of C2PS model in the smart transport arena can be used as a template for other CPS domains such as smart home, smart office, smart agriculture, smart waste management, smart buildings etc.

The C2PS reference model can be very useful for greater level of integration among different CPS domains. This integration requires a common language, which can be formulated from a upper level meta ontology. Later, domain specific ontologies can be developed to facilitate these integrations. As a result, a true Smart city can be formed, where data can be accessed across domains seamlessly.

The VCPS implementation can also incorporate people-to-X communications. As people are key users of the VCPS data, future works can explore this integration with the other VCPS subsystems. There is ample scope to improve the *tNote* data structure, so that it can be very handy for the data mining applications. The SIoV data workload models might be improved by introducing probabilistic models. Also, in case of the SIoV scenarios, integration of the continuous and the discrete time can be an interesting problem. Furthermore, implementations and deployment of additional adaptive algorithms for the VCPS subsystems can be a productive study.

Continuous development of the C2PS platform, subsequent real life deployments, data collection of the vehicular scenarios are important future works. The platform can also be modified to collect other ubiquitous sensory data (e.g. pollution data). Large scale deployment of the platform requires heavy cloud infrastructure availability. We can also compare the performance of the platform for cloud computing and fog computing type of deployments. Continuous development of the multi-sensory VCPS dataset is also an important future work that can benefit the related research community. Addition of more dataset analysis tools (e.g. MATLAB) is also useful.

Open sourcing the C2PS platform can be helpful for the research community. We can use the platform to build more safety, efficiency, comfort and entertainment applications. C2PS and its vehicular domain, VCPS, are in their early stage of development. Participation from the research community to improve the C2PS application development platform will be beneficial for both the research community and the industry.

This dissertation is a small step towards the development and deployment of the vehicular cyber-physical system applications. We hope in the coming time, we will see true adoption of various C2PS based Smart City applications.

References

[1] Ian Ku, You Lu, Mario Gerla, Francesco Ongaro, Rafael L Gomes, and Eduardo Cerqueira. Towards software-defined vanet: Architecture and services. In *Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean*, pages 103–110. IEEE, 2014.

[2] Yunxin Jeff Li. An overview of the dsrc/wave technology. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 544–558. Springer, 2010.

[3] Mukesh Saini, Abdulhameed Alelaiwi, and Abdulmotaleb El Saddik. How close are we to realizing a pragmatic vanet solution? a meta-survey. *ACM Computing Surveys (CSUR)*, 48(2):29, 2015.

[4] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[5] Dongyao Jia, Kejie Lu, Jianping Wang, Xiang Zhang, and Xuemin Shen. A survey on platoon-based vehicular cyber-physical systems. 2015.

[6] Annalisa Cocchia. Smart and digital city: a systematic literature review. In *Smart city*, pages 13–43. Springer, 2014.

[7] Department of Economic and Social Affairs. World urbanization prospects: The 2014 revision. United Nations, 2014.

[8] Machine-to-machine communications (m2m); impact of smart city activity on iot environment. European Telecommunications Standards Institute (ETSI), 2015.

[9] Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1028–1031. IEEE, 2011.

[10] Mahmoud Al-Hader, Ahmad Rodzi, Abdul Rashid Sharif, and Noordin Ahmad. Smart city components architicture. In *Computational Intelligence, Modelling and Simulation, 2009. CSSim'09. International Conference on*, pages 93–97. IEEE, 2009.

[11] Melodie Warner. General motors sales up in december. 2013.

[12] E. L. George. The us economy and monetary policy. 2013.

[13] Intelligent Vehicle Initiative. Saving lives through advanced vehicle safety technology. *US Department of Transportation, http://www. its. dot. gov*, 2005.

[14] NHTSA. National highway traffic safety administration. usdot. *NHTSA press*, 2007.

[15] Intelligent transport systems (its); vehicular communications; basic set of applications; definitions. European Telecommunications Standards Institute (ETSI), 2009.

[16] Steven Plotkin, Thomas Stephens, and Walter McManus. Vehicle technology deployment pathways: An examination of timing and investment constraints. Technical report, Transportation Energy Futures Series, Department of Energy, 2013.

[17] Markus Schütz and Klaus Dietmayer. A flexible environment perception framework for advanced driver assistance systems. In *Advanced Microsystems for Automotive Applications 2013*, pages 21–29. Springer, 2013.

[18] Yasser L Morgan. Notes on dsrc & wave standards suite: Its architecture, design, and characteristics. *Communications Surveys & Tutorials, IEEE*, 12(4):504–518, 2010.

[19] Al-Sakib Khan Pathan. *Securing Cyber-physical Systems*. CRC Press, 2015.

[20] Tarik Taleb, Ehssan Sakhaee, Abbas Jamalipour, Kazuo Hashimoto, Nei Kato, and Yoshiaki Nemoto. A stable routing protocol to support its services in vanet networks. *Vehicular Technology, IEEE Transactions on*, 56(6):3337–3347, 2007.

[21] Sunder Aditya Rao, M Pai, Mounir Boussedjra, and Joseph Mouzna. Gpsr-l: Greedy perimeter stateless routing with lifetime for vanets. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 299–304. IEEE, 2008.

[22] Yufeng Chen, Zhengtao Xiang, Wei Jian, and Weirong Jiang. A cross-layer aomdv routing protocol for v2v communication in urban vanet. In *Mobile Ad-hoc and Sensor Networks, 2009. MSN'09. 5th International Conference on*, pages 353–359. IEEE, 2009.

[23] Ram Shringar Raw and DK Lobiyal. E-dir: a directional routing protocol for vanets in a city traffic environment. *International Journal of Information and Communication Technology*, 3(3):242–257, 2011.

[24] Aneel Rahim, Zeeshan Shafi Khan, Fahad T Bin Muhaya, Muhammad Sher, and Tai-Hoon Kim. Sensor based framework for secure multimedia communication in vanet. *Sensors*, 10(11):10146–10154, 2010.

[25] M. Nitti, R. Girau, A. Floris, and L. Atzori. On adding the social dimension to the internet of vehicles: Friendship and middleware. In *Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on*, pages 134–138, May 2014.

[26] Fangchun Yang, Shangguang Wang, Jinglin Li, Zhihan Liu, and Qibo Sun. An overview of internet of vehicles. *Communications, China*, 11(10):1–15, 2014.

[27] Yi Zhe Zhou. Research on key technology of multi-channel transmission about mac layer safety information in internet of vehicles. In *Applied Mechanics and Materials*, volume 556, pages 6320–6323. Trans Tech Publ, 2014.

[28] Liviu Miclea and Teodora Sanislav. About dependability in cyber-physical systems. In *Design & Test Symposium (EWDTS), 2011 9th East-West*, pages 17–21. IEEE, 2011.

[29] Siddhartha Kumar Khaitan and James D McCalley. Design techniques and applications of cyberphysical systems: A survey. *Systems Journal, IEEE*, 9(2):350–365, 2015.

[30] Hassan Abid, Luong Thi Thu Phuong, Jin Wang, Sungyoung Lee, and Saad Qaisar. V-cloud: vehicular cyber-physical systems and cloud computing. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, page 165. ACM, 2011.

[31] Wu He, Gongjun Yan, and Li Da Xu. Developing vehicular data cloud services in the iot environment. *IEEE Transactions on Industrial Informatics*, 10(2):1587–1595, 2014.

[32] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.

[33] Jiafu Wan, Daqiang Zhang, Yantao Sun, Kai Lin, Caifeng Zou, and Hu Cai. Vcmia: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. *Mobile Networks and Applications*, 19(2):153–160, 2014.

[34] Salim Bitam, Abdelhamid Mellouk, and Sherali Zeadally. Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *Wireless Communications, IEEE*, 22(1):96–102, 2015.

[35] Huansheng Ning and Ziou Wang. Future internet of things architecture: Like mankind neural system or social organization framework? *Communications Letters, IEEE*, 15(4):461–463, April 2011.

[36] Matthias Kranz, Luis Roalter, and Florian Michahelles. Things that twitter: social networks and the internet of things. In *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, 2010.

[37] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Siot: Giving a social structure to the internet of things. *Communications Letters, IEEE*, 15(11):1193–1195, 2011.

[38] Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, 2012.

[39] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Sensing as a service model for smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies*, 25(1):81–93, 2014.

[40] M Tamer Özsu and Patrick Valduriez. *Principles of distributed database systems*. Springer Science & Business Media, 2011.

[41] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 44–51. Ieee, 2009.

[42] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[43] Edward A Lee. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.

[44] Ragunathan Raj Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference*, pages 731–736. ACM, 2010.

[45] Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. A survey of cyber-physical systems. In *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, pages 1–6. IEEE, 2011.

[46] National Science Foundation (NSF). Cyber-physical systems (cps) nsf 11-516, 2013.

[47] Jaehoon Paul Jeong and Eunseok Lee. Vehicular cyber-physical systems for smart road networks. *KICS Information and Communications Magazine*, 31(3):103–116, 2014.

[48] D. Guinard, M. Fischer, and V. Trifa. Sharing using social networks in a composable web of things. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 702–707, March 2010.

[49] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing*, pages 116–122. Springer, 2001.

[50] Andreas Barthels, Joachim Fröschl, Hans-Ulrich Michel, and Uwe Baumgarten. An architecture for power management in automotive systems. In *Architecture of Computing Systems–ARCS 2012*, pages 63–73. Springer, 2012.

[51] Maximilian Engelsberger and Thomas Greiner. Software architecture for cyber-physical control systems with flexible application of the software-as-a-service and on-premises model. In *Industrial Technology (ICIT), 2015 IEEE International Conference on*, pages 1544–1549. IEEE, 2015.

[52] Jiang Wan, Arquimedes Canedo, Al Faruque, and Mohammad Abdullah. Functional model-based design methodology for automotive cyber-physical systems. 2014.

[53] A. M. Vegni and V. Loscr. A survey on vehicular social networks. *IEEE Communications Surveys Tutorials*, 17(4):2397–2419, Fourthquarter 2015.

[54] Stephen Smaldone, Lu Han, Pravin Shankar, and Liviu Iftode. Roadspeak: enabling voice chat on roadways using vehicular social networks. In *1st Workshop on Social Network Systems*, pages 43–48. ACM, 2008.

[55] Xiping Hu, Weihong Wang, and Victor Leung. Vssa: a service-oriented vehicular social-networking platform for transportation efficiency. In *The second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 31–38. ACM, 2012.

[56] Mohamed Eltoweissy, Stephan Olariu, and Mohamed Younis. Towards autonomous vehicular clouds. In *Ad hoc networks*, pages 1–16. Springer, 2010.

[57] Xiping Hu, Victor Leung, Kevin Garmen Li, Edmond Kong, Haochen Zhang, Nambiar Shruti Surendrakumar, and Peyman TalebiFard. Social drive: a crowdsourcing-based vehicular social networking system for green transportation. In *Third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 85–92. ACM, 2013.

[58] Rifaqat Hussain, Junggab Son, Hasoo Eun, Sangjin Kim, and Heekuck Oh. Rethinking vehicular communications: Merging vanet with cloud computing. In *Cloud*

Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pages 606–609. IEEE, 2012.

[59] Hesham El-Sayed, Liren Zhang, Yasser Hawas, and Hadeel El Kassabi. A histogram-based model for road traffic characterization in vanet. In *Internet of Vehicles–Technologies and Services*, pages 42–55. Springer, 2014.

[60] Omar Chakroun and Soumaya Cherkaoui. Rmds: Relevance-based messages dissemination scheme for 802.11 p vanets. In *Communications (ICC), 2014 IEEE International Conference on*, pages 568–573. IEEE, 2014.

[61] Gongjun Yan, Weiming Yang, Danda B Rawat, and Stephan Olariu. Smartparking: A secure and intelligent parking system. *Intelligent Transportation Systems Magazine, IEEE*, 3(1):18–30, 2011.

[62] Saleh Yousefi, Eitan Altman, Rachid El-Azouzi, and Mahmood Fathy. Analytical model for connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(6):3341, 2008.

[63] Dmytro Rud, Andreas Schmietendorf, and Reiner Dumke. Performance modeling of ws-bpel-based web service compositions. In *Services Computing Workshops, 2006. SCW'06. IEEE*, pages 140–147. IEEE, 2006.

[64] Roberta Fracchia and Michela Meo. Analysis and design of warning delivery service in intervehicular networks. *Mobile Computing, IEEE Transactions on*, 7(7):832–845, 2008.

[65] Feng Xia et al. Performance modeling methodology of emergency dissemination algorithms for vehicular ad-hoc networks. In *2010 7th International Symposium on Communication Systems, Networks&Digital Signal Processing (CSNDSP 2010)*, pages 397–400, 2010.

[66] Mehdi Khabazian, Sonia Aissa, and Mustafa Mehmet-Ali. Performance modeling of message dissemination in vehicular ad hoc networks with priority. *Selected Areas in Communications, IEEE Journal on*, 29(1):61–71, 2011.

[67] Antonio M Ortiz, DH Ali, Soochang Park, Son N Han, and Noel Crespi. The cluster between internet of things and social networks: Review and research challenges. 1(3):206–215, 2014.

[68] Kate Carruthers. Internet of things and beyond: Cyber-physical systems, 2016.

[69] Klaus Schwab. The fourth industrial revolution: what it means and how to respond, 2015.

[70] Kazi Masudul Alam, Mukesh Saini, and Abdulmotaleb El Saddik. tnote: A social network of vehicles under internet of things. In *Internet of Vehicles–Technologies and Services*, pages 227–236. Springer International Publishing, 2014.

[71] Kazi Masudul Alam, Mukesh Saini, Dewan T Ahmed, and Abdulmotaleb El Saddik. Vedi: A vehicular crowd-sourced video social network for vanets. In *IEEE 39th Conference on Local Computer Networks (LCN)*. 2014.

[72] Kazi Masudul Alam, Mukesh Saini, and Abdulmotaleb El Saddik. Toward social internet of vehicles: Concept, architecture, and applications. *Access, IEEE*, 3:343–357, 2015.

[73] Kazi Masudul Alam, Mukesh Saini, and Abdulmotaleb El Saddik. Workload model based dynamic adaptation of social internet of vehicles. *Sensors*, 15(9):23262–23285, 2015.

[74] Kazi Masudul Alam, Alex Sopena, and Abdulmotaleb El Saddik. Design and development of a cloud based cyber-physical architecture for the internet-of-things. In

Multimedia (ISM), 2015 IEEE International Symposium on, pages 387–392. IEEE, 2015.

[75] Wei Wang, Suparna De, Ralf Toenjes, Eike Reetz, and Klaus Moessner. A comprehensive ontology for knowledge representation in the internet of things. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1793–1798. IEEE, 2012.

[76] Sara Hachem, Thiago Teixeira, and Valérie Issarny. Ontologies for the internet of things. In *Proceedings of the 8th Middleware Doctoral Symposium*, page 3. ACM, 2011.

[77] Edward Ashford Lee and Sanjit Arunkumar Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. Lee & Seshia, 2011.

[78] Andreas Barthels, Florian Ruf, Gregor Walla, Joachim Fröschl, Hans-Ulrich Michel, and Uwe Baumgarten. A model for sequence based power management in cyber physical systems. In *Information and Communication on Technology for the Fight against Global Warming*, pages 87–101. Springer, 2011.

[79] Danda B Rawat, Joel JPC Rodrigues, and Ivan Stojmenovic. *Cyber-Physical Systems: From Theory to Practice*. CRC Press, 2015.

[80] Panu Korpipää, Miika Koskinen, Johannes Peltola, Satu-Marja Mäkelä, and Tapio Seppänen. Bayesian approach to sensor-based context awareness. *Personal and Ubiquitous Computing*, 7(2):113–124, 2003.

[81] Mark Weiser. Reasoning about uncertain contexts in pervasive computing environments. 2004.

[82] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International*

Conference on Fuzzy Systems and Knowledge Discovery, pages 970–979. Springer, 2006.

[83] Irad Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 2007.

[84] Judea Pearl. Bayesian networks. *Department of Statistics, UCLA*, 2011.

[85] Chuen Chien Lee. Fuzzy logic in control systems: fuzzy logic controller. ii. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):419–435, 1990.

[86] Jin Zhao and Bimal K Bose. Evaluation of membership functions for fuzzy logic controlled induction motor drive. In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, volume 1, pages 229–234. IEEE, 2002.

[87] Sameena Naaz, Afshar Alam, and Ranjit Biswas. Effect of different defuzzification methods in a fuzzy based load balancing application. *IJCSI-International Journal of Computer Science Issues*, 8(5), 2011.

[88] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. Mobile fog: A programming model for large-scale applications on the internet of things. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pages 15–20. ACM, 2013.

[89] Ghassan Samara. Dynamic safety message power control in vanet using pso. *arXiv preprint arXiv:1401.2405*, 2014.

[90] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys & Tutorials, IEEE*, 13(4):584–616, 2011.

[91] Stephan Haller. The things in the internet of things. In *Internet of Things Conference*, 2010.

[92] Alexandru Serbanati, Carlo Maria Medaglia, and Ugo Biader Ceipidor. Building blocks of the internet of things: State of the art and beyond. *Deploying RFID-Challenges, Solutions, and Open Issues, C. Turcu, Ed., InTech*, 2011.

[93] Hassnaa Moustafa and Yan Zhang. *Vehicular networks: techniques, standards, and applications*. Auerbach publications, 2009.

[94] Hasnaa Moustafa, Gilles Bourdon, and Yvon Gourhant. Providing authentication and access control in vehicular network environment. In *Security and Privacy in Dynamic Environments*, pages 62–73. Springer, 2006.

[95] Michael Feld and Christian Müller. The automotive ontology: managing knowledge inside the vehicle and sharing it between cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 79–86. ACM, 2011.

[96] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J Martinez, J-C Cano, Carlos T Calafate, and Pietro Manzoni. Caova: A car accident ontology for vanets. In *IEEE Wireless Communications and Networking Conference*, pages 1864–1869, 2012.

[97] João Ferreira, Francesco Furini, and Nuno Silva. Vco–vehicle corporate ontology. In *2nd Conf. Ibérica de Sistemas e Tecnologias de Informação*, 2007.

[98] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo-simulation of urban mobility-an overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 55–60, 2011.

[99] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[100] Yasser L Morgan. Managing dsrc and wave standards operations in a v2v scenario. *International Journal of Vehicular Technology*, 2010, 2010.

[101] Seyed Vahid Hosseinioun, Hussein Al-Osman, and AE Saddik. Employing sensors and services fusion to detect and assess driving events. In *2015 IEEE International Symposium on Multimedia (ISM)*, pages 395–398. IEEE, 2015.

[102] Kazi Masudul Alam, Mohammed Bin Hariz, Seyed Vahid Hosseinioun, Mukesh Saini, and Abdulmotaleb El Saddik. Mudva: A multi-sensory dataset for the vehicular cps applications. In *Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop on*, pages 1–6. IEEE, 2016.

APPENDICES

Appendix A

SIoV Graphical User Interfaces

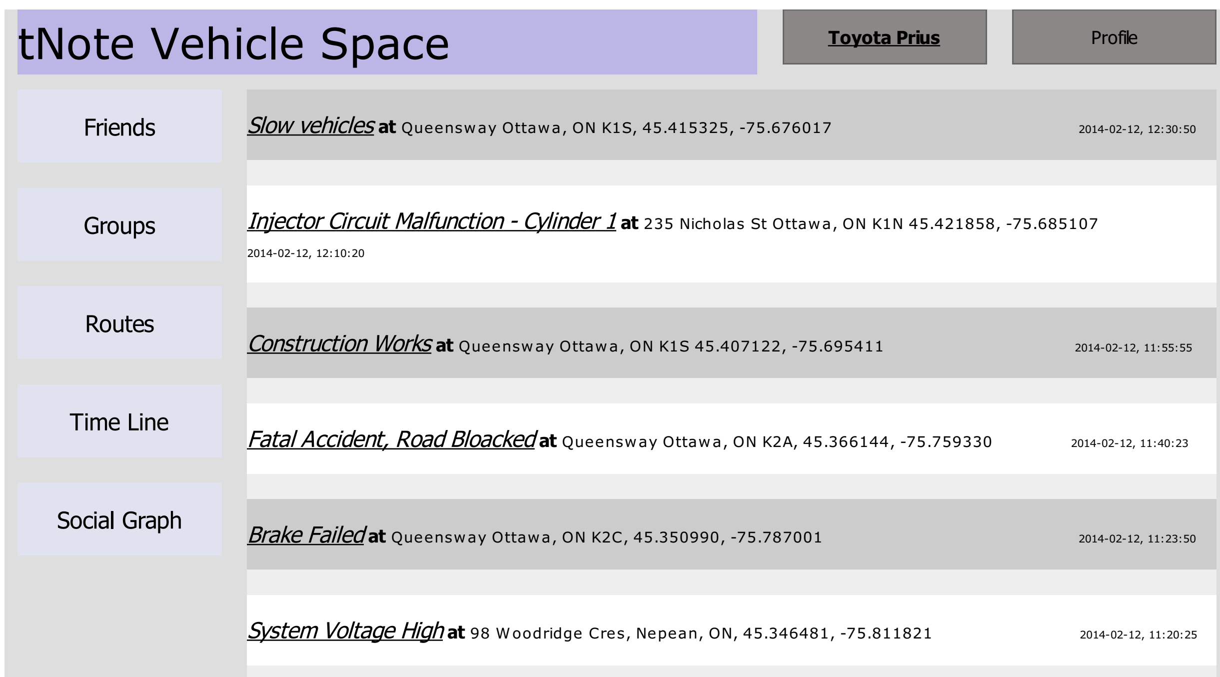


Figure A.1: Home view for vehicle `Toyota Prius`

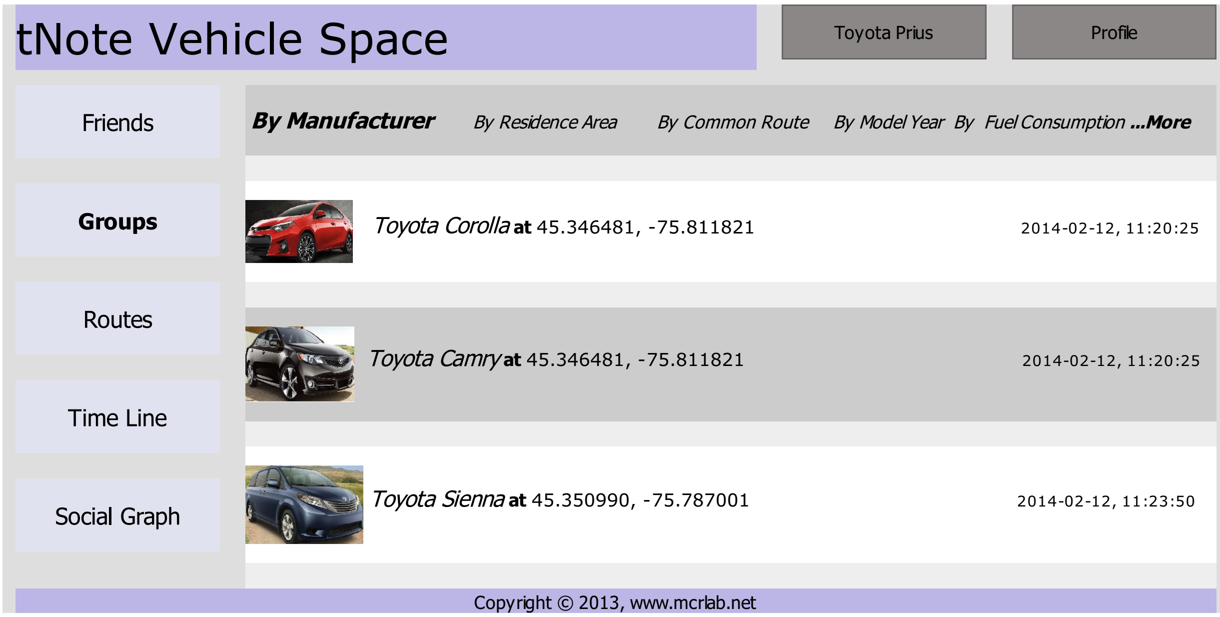


Figure A.2: Groups view based on manufacturer category

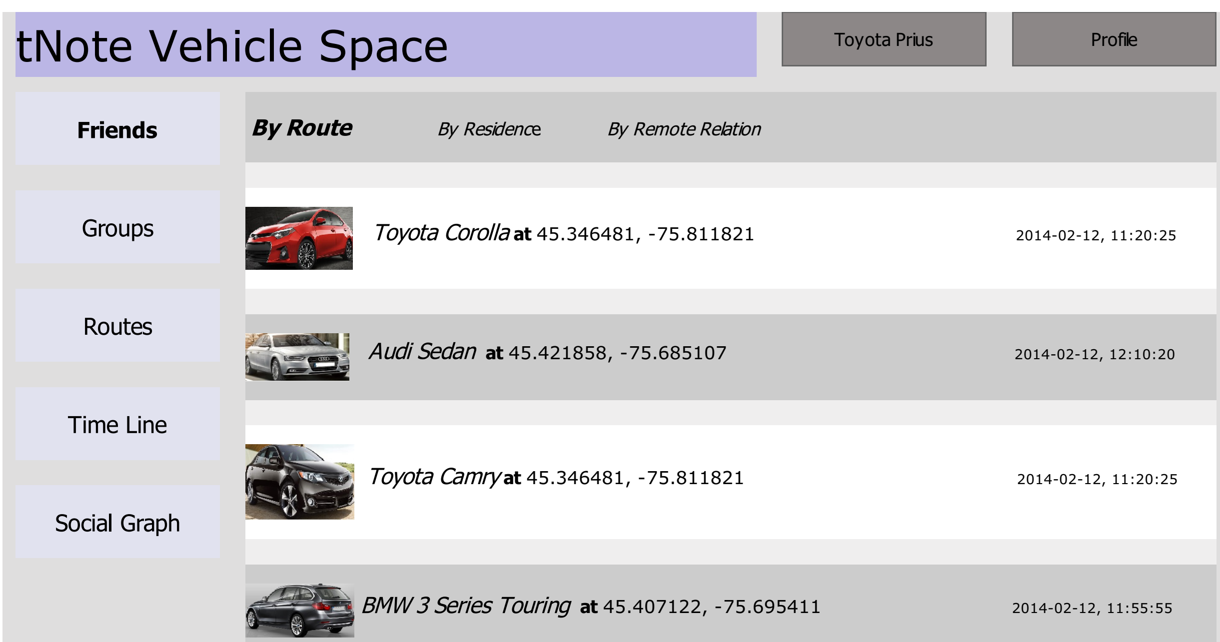


Figure A.3: Friends view based on routes category

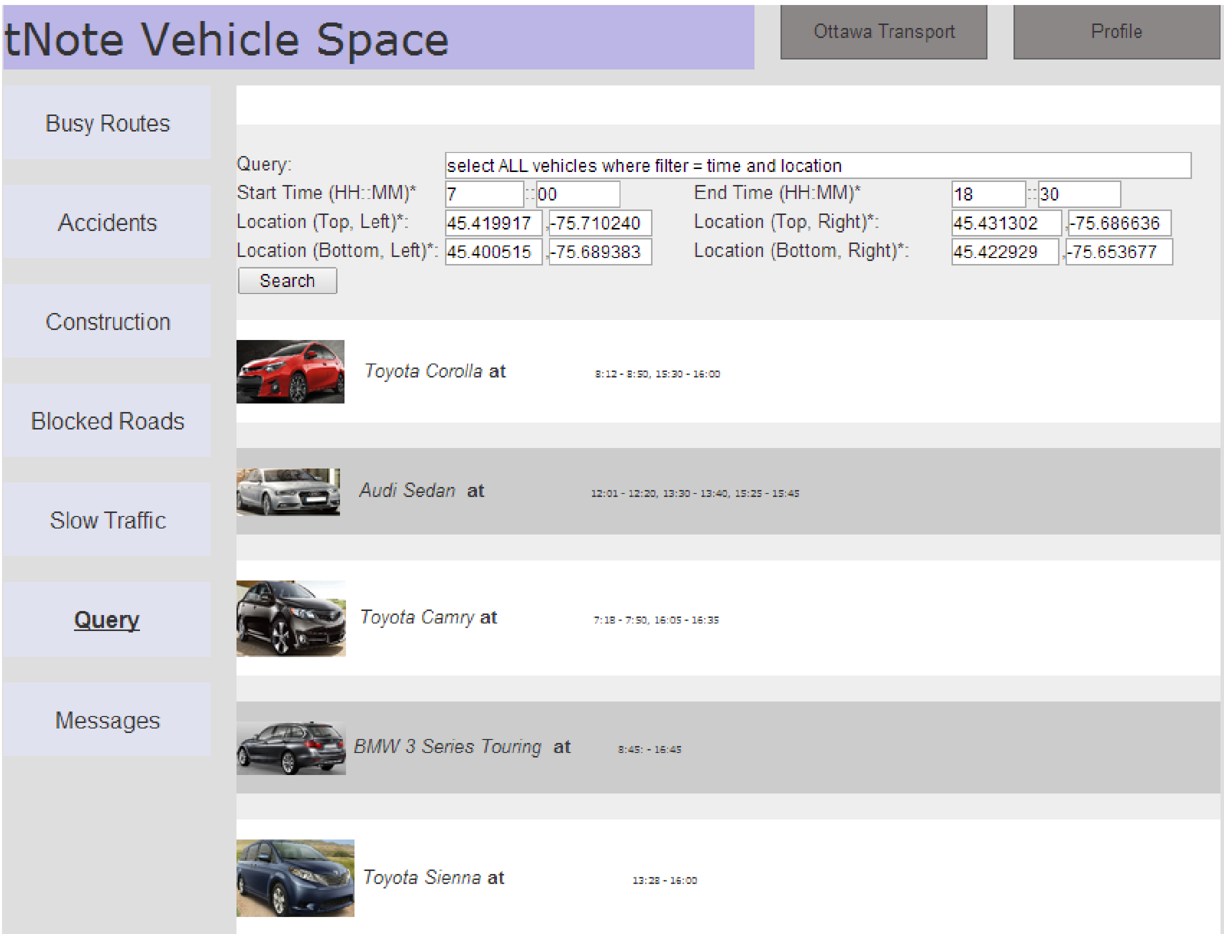


Figure A.4: Transport authority query view