Collision avoidance and Drone surveillance using Thread protocol in V2V and V2I communications

Rajas Chitanvis, Niranjan Ravi, Tanmay Zantye and Mohamed El-Sharkawy
IoT Collaboratory IUPUI, Department of Electrical and Computer Engineering
Purdue School of Engineering and Technology Indianapolis
rachitan@iu.edu ravin@iu.edu tzantye@iu.edu melshark@iupui.edu

Abstract—**According to the World Health Organizations (WHO) report nearly 1.25 million people die in road accidents every year. This creates a need for Advanced Driver Assist Systems (ADAS) which can ensure safe travel. To tackle the above challenge in existing the ADAS, Intra-vehicular communications (V2V) and vehicle to infrastructure communications (V2I) has been one of the predominant research topics nowadays due to the rapid growth of automobile industries and ideology of producing autonomous cars in the near future. The key feature of V2V communication is vehicle to vehicle collision detection by transmitting information like vehicle speed and position of a vehicle to other vehicles in the same location using wireless sensor networks (WSN). On the other hand, Unmanned Aerial Vehicle (UAV) systems are growing at a rapid rate in various aspects of life including dispatch of medicines and undergo video surveillance during an emergency due to less air traffic. This paper demonstrates the practice of integrating V2V communication with Thread, one of the low power WSN for data transmission, to initiate adaptive cruise control in a vehicle during a crisis. Also, UAV systems are employed as a part of V2I system to provide aerial view video surveillance if any accident occurs.**

Index Terms—**V2V, V2I, MBDT, Thread, IEEE 802.15.4, IBM Watson, PX4, UAV.**

I. Introduction

The automobile industry has been growing at a rapid phase during the last decade. This has led to the production of more advanced and high-speed vehicles at a reduced cost. But with this development many challenges arose to address the safety issues of people travelling in the car and the pedestrians as well. Implementing advanced security features and improved enhancements to existing vehicle architectures has been one of the hot topics for researchers. This paved way for the development of V2V and V2I systems. V2V is a communication system that establishes a wireless communication between the vehicles and ensuring the safety of passengers by reducing the collisions. This could prevent hazards like vehicle crashing or unexpected crashes. This system can further expand by connecting all the vehicles in a particular zone, like a traffic location or could be a community, in a single wireless network which has no single point of failure. As a result of this integration, all the vehicles would be alerted in the event of a crash or any hazardous situation. On the other hand, vehicle to infrastructure communication is a wireless exchange of data between vehicles and infrastructure. V2I communication is an embedded system which consists of hardware, software and firmware systems. The data from infrastructures could be transmitted to anywhere in the world with the help of cloud platforms. By sharing these data, a wide range of safety, mobility and environmental benefits can be employed. There are few challenges in implementing V2V and V2I systems into existing vehicle systems like the range of wireless networks, establishing a network without failure and transmission of data from infrastructure to destined locations, and security of data transmission during this tenure. Thread is a secure wireless networking protocol which is based on IPv6 (most recent version of Internet Protocol) which expands the range WSN to an infinite number of involved devices to exchange data between them. For this reason, the Thread is chosen to demonstrate V2V and V2I communications. Transmission of data is done efficiently using IBM cloud platform which also aids in encryption of data using ECC cryptography. As an additional and efficient feature, the usage of UAV systems during vehicle accidents by implementing Thread is discussed in this paper. This could offer a foolproof system with a quick response time during an event of crash.

II. RELATED STUDY ON VEHICLE COMMUNICATION

A. *Vehicle-To-Vehicle Communication*

Vehicle-to-Vehicle communication is the ability of a vehicle to share its information wirelessly to other surrounding vehicles. Generally, the information is about vehicle location, heading direction and speed. V2V communication comprises of sending and receiving omni-directional messages which is used to determine potential accidents/crashes/threats. This technology is being utilized to alert the drive using visual and audio means. V2V communication is an enhancement to currently available crash avoidance systems which includes Camera, Lidar, Radar and Ultrasonic sensors to detect collision threats. This technology is focused to improve vehicular safety and help save lives[1].

B. *Vehicle-To-Infrastructure Communication*

Vehicle to Infrastructure Communication is a vastly growing communication protocol, widely used in the field of Intelligent Transportation Systems (ITS) which allows a bidirectional wireless data transfer between the vehicle and infrastructures such as road signs, traffic signals and lane markings. The sensors used in road infrastructures send important information about traffic congestion, speed management and warnings for

Chitanvis, R., Ravi, N., Zantye, T., & El-Sharkawy, M. (2019). Collision avoidance and Drone surveillance using Thread protocol in V2V and V2I communications. 2019 IEEE National Aerospace and Electronics Conference (NAECON), 406–411. https://doi.org/10.1109/NAECON46414.2019.9058170

hazardous situations such as accidents, which can be sent to a vehicle in order to improve the safety. On the other hand, data such as speed, acceleration and location of the vehicle is sent to road infrastructure, to analyse the impact of the vehicle in overall traffic safety. This information can also be used in applications such as dynamic traffic control, parking control and highway toll control. V2I infrastructure uses the Dedicated Short Range Communication (DSRC) for wireless data transfer between the vehicle and on road infrastructures. The DSRC consists of an on-board unit (OBU) and a roadside unit (RSU). The OBU performs real-time communication in between the vehicle and not only road side infrastructure, but also other vehicles by collecting data from multiple sensors such as GPS, accelerometer and storage unit. The RSU is placed at intersections and interchanges which allows necessary information to be passed to vehicles and other devices. Protocols such as Bluetooth, Thread and WiFi can also be used for wireless data transfer. Vehicle to Infrastructure communication promises significant developments in the fields of vehicle safety and traffic congestion[4].

III. RELATED STUDY ON WIRELESS SENSOR NETWORKS

Wireless sensor networks is a network of sensors that communicates the information gathered through wireless links. It consists of base stations and a number of wireless sensors which monitor physical conditions such as sound, temperature, light, pressure etc. The data is relayed through multiple nodes and is connected with other networks like the internet using a common gateway. WSN supports star, tree and the mesh topologies using radio communication networks. Mesh topologies allows data transmission from one node to another when both nodes are within radio transmission range. If the node wants to transmit data to another node, which is not in its radio transmission range, it requires an intermediate node to relay the message to the desired node. Mobile WSN (MWSN) is a network of mobile sensors which can move on its own. MWSN are more versatile than a static sensor network as they have the advantage of moving and covering more area[6].

IV. THREAD PROTOCOL

Thread is a secure wireless networking protocol which is based on IPv6. Thread stack is built on the collection of existing Institute of Electrical and Electronics Engineers (IEEE) and Internet Engineering Task Force (IETF) standards which supports many different topologies and it also ensures that only authorized devices are added to the network. Thread stack has no standard application layer which enables developers to use any necessary application layer. Thread networks are encrypted using Advanced Encryption Standard (AES). This provides security at the network layer which also can also be added to the application layer. Thread stack is designed to provide no single point of failure in the mesh network even if a node is lost it, will be replaced without affecting the ongoing operations of the network. Thread uses User Datagram Protocol (UDP) for network layer on top of Internet Protocol (IP) routing and IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN), it also utilises IEEE 802.15.4 Media Access Control (MAC) and Physical (PHY) wireless specifications providing mesh network with maximum speed of 250 Kbps in 2.4GHz band[2].

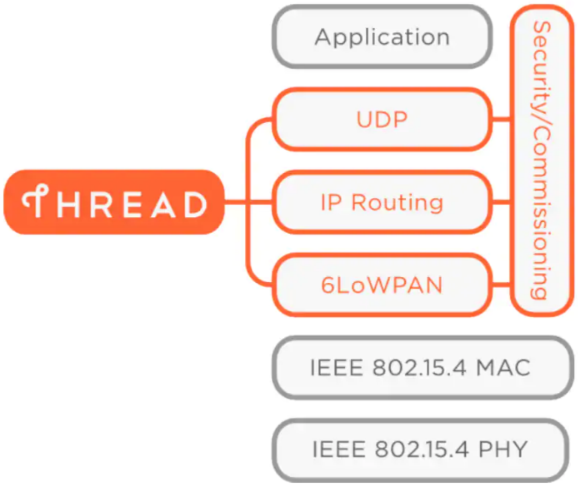

Fig. 1. Thread Network Stack[11]

A. Border Router

A Border Router provides an interface between 802.15.4 networks and adjacent physical layers such as Wi-Fi, Ethernet etc. This enables the thread mesh network to share its data with the internet/cloud using a border router. There is at least one border router in the mesh network and it is possible to have many.

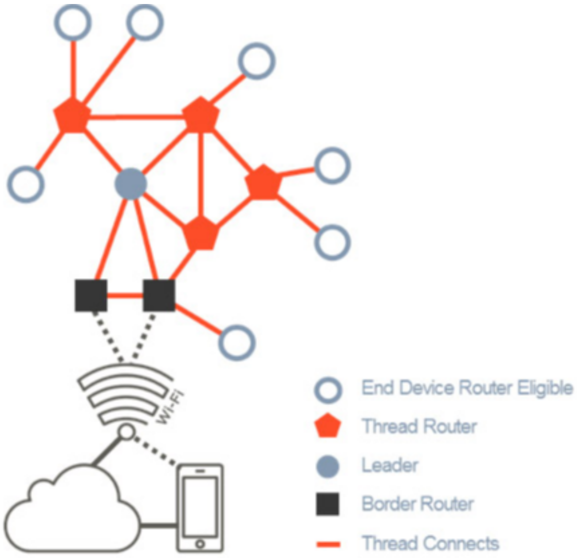

Fig. 2. Thread Mesh Network[12]

A leader manages the thread mesh network by assigning router IDs and accepting requests from router-eligible end devices (REEDs) to become a Thread Router. It uses CoAP (Constrained Application Protocol) to assign roles to all nodes in the Thread mesh network. According to the topology of the mesh network it assigns roles such as Thread Routers and End Devices. A Leader governs a registry which includes assigned router IDs and this information is also shared with Thread Routers. So, if the Leader loses connectivity with the Thread mesh network, another Thread Router is elected as a Leader without user intervention[2].

B. *Thread Router*

A Thread Router provides joining, security and routing services to devices in the mesh network and to the devices wanting to join the network. While Thread Routers are not allowed to sleep, they can downgrade their functionality and become REEDs, considering the topology and Leader allows them to downgrade[2].

C. *Router Eligible End Devices(REEDs)*

REED is a router eligible end device that has a capability to become the Thread Router or even a Leader. Because of mesh network topology, REEDs do not act as routers and do not provide joining or security services to the network devices[2].

V. Cloud Computing

The recent advancements in the field of computing lead to a rise in digital content which created a need for cloud platform. Cloud computing provides a scalable platform for researchers and developers. Cloud computing not only includes services which are sent through the internet but also at a hardware level which is responsible for sending those services. It solves the problems faced by hardware devices like limited storage capacity and absence of data visualization. Cloud computing extends its services to different levels of infrastructure from IoT devices to database servers. This integration of cloud with IoT has enabled researchers to store unlimited amount of real-time data and interpret it by using complex algorithms which would be difficult to implement at hardware levels. The other advantage of using cloud framework with IoT devices is that the data can be handled remotely by any individual through the cloud. The range of data transmission is a constraint faced by IoT devices in recent times. By this integration, data can be transferred between any number of devices simultaneously at an exponential rate. Cloud provides better collaboration acting as a pathway for the data to travel and by reducing the overall cost of IoT devices[8].

VI. Elliptic-curve cryptography(ECC)

In the era of internet, digital content is rising at a rapid rate. This brings huge risk for privacy and to sensitive data which is being shared. The above security threats can be overcome by using cryptographic algorithms which provide confidentiality, integrity and authentication during data exchange. ECC and Rivest Shamir Adleman (RSA) are widely used cryptographic algorithms by researchers. The former is employed in many resource constrained IoT devices (between ram 10kb and 100). An adequate level of security is needed since data exchange would happen upon integrating IoT with other WSN or cloud platforms. Each device in the system would be deployed with a unique pre-shared key[9]. This key serves as a master key for initial setup and authentication between the devices in the network. For continuous message encryption between trusted devices, Integrated Encryption Scheme (IES) is employed which would harness the speed of the algorithm during large amount of data transfer by avoiding repeated key exchange for each unit of data transfer. Elliptic Curve Digital Signature Algorithm (ECDSA) signatures are integer keypairs and the complexity of the algorithm increases with addition of keys which would increase the security of the applications as well[10].

VII. Unmanned Aerial Vehicle (UAV)

A. *Pixhawk 4*

Pixhawk is the hardware standard which supports open source flight stacks such as PX4 and ArduPilot[5]. Pixhawks flight management unit (FMU) processor is based on STM32F765 Cortex-M7 and the IO processor is based on STM32F100 Cortex-M3. It is embedded with on-board accelerometer, gyroscope, magnetometer and barometer.The Pixhawk provides I-Squared Communication (I2C), Serial Peripheral Interface (SPI), Controller Area Network (CAN), Pulse Position Modulation (PPM) and DSM interfaces with R/C transceiver, sensors, telemetry, GPS and a companion computer. It provides 8 IO PWM outputs and 8 FMU PWM outputs which are used to control Brushless DC (BLDC) and other actuators such as Servo motors[3].

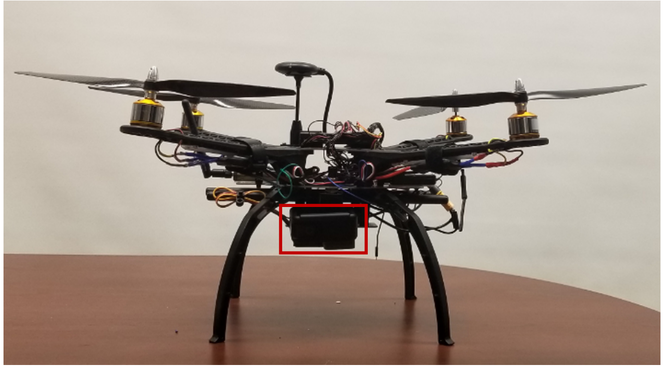

Fig. 3. UAV system with a Aerial View Camera

B. *PX4*

PX4 is a open source flight stack and powers all kind of vehicles from drones from ground vehicles to submarine. Its system architecture allows to modify the flight stack and middleware to add new flight modes and new air frames. PX4 supports sensors and actuators like camera, LiDAR, Servo motor etc. It supports MAVLink which is a highly efficient, lightweight and fast communication protocol. PX4 provides flexible simulation framework like AirSim, Robot Operating System (ROS) Gazebo Simulator, jMAVSim and X-Plane. This flight stack can be build on a variety of autopilot hardware like Pixhawk series, Nxphlite, Snapdragon Flight, Intel Aero, Parrot Bebop, Raspberry Pi etc[3].

VIII. Model Based Design Toolbox (MBDT)

The Model Based Design Toolbox is an integrated development environment designed by NXP to configure and generate all mandatory software automatically on various Micro-controllers (MCU). It performs vital tasks such as initialization of routines and device drivers in order to execute the selected applications such as motor control algorithms and communication protocols like CAN,Universal Asynchronous Receiver/Transmitter (UART), I2C, etc. The toolbox provides pre-existing examples blocks for various applications such as motor control, communication protocols and sensor-based applications which are integrated with simulink embedded target for various MCUs. The toolbox provides drag-drop programming, graphical-based architecture and Real Time Operating System (FreeRTOS) integration which enables fast designing, verification and testing on real targets for development in Matlab environment[7].

IX. System Design

The system design consists of two miniature car models. Each car model would consist of a Lidar Lite V3HP connected to NXP's FRDM k64F board using I2C communication protocol. One portion of K64F board is in-turn connected to S32K144 [ISO Functional safety approved product] which is embedded with Motor GD Devkit specially designed for controlling Permanent Magnet Synchronous Motor (PMSM) motors. A Linux motor which has a rated voltage of about 24V and a rated speed of 4000 RPM is fixed to the GD Devkit. The other portion of K64F is attached with the FRDM KW41Z which supports thread radio Modules. These Thread networks act as thread router[1] with a range of 100m. In addition to the above system, we have added another thread network device (KW41Z) which acts as border router with NXPs 71ch, a security chip for IoT applications to encrypt all the data which is being received onto it and uploads it to the IBM cloud platform through Wi-Fi or ethernet connectivity. This setup can be placed in any infrastructure like traffic signals, buildings adjacent to city traffic or even gas stations. In a remote location, a drone is placed with a flight controller module called Pixhawk. Pixhawk is an autopilot module running on PX4 flight stack which allows the drone to plan autonomous missions. A raspberry pi and a vertical camera for aerial view surveillance is attached to it. Raspberry pi would subscribe the data from IBM cloud every second.

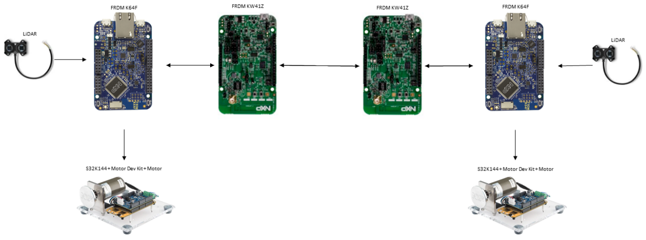

Fig. 4. V2V System Design

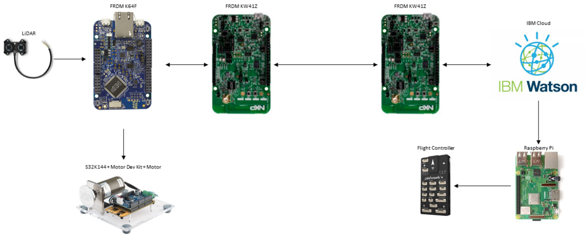

Fig. 5. V2I System Design

X. Hardware and software requirements

- Lidar lite v3HP.
- NXP FRDM K64F boards.
- NXPs FRDM KW41Z boards.
- NXPs S32K144 with Devkit Motor GD.
- NXPs A71Ch-Plug and Trust for IoT.
- UAV system with Pixhawk, Raspberry pi and a servo motor.
- jMAVSim for UAV systems.
- Motor Based Design ToolBox.

XI. Implementation

Case1: Collision avoidance system using adaptive cruise control in vehicle-to vehicle communication. As discussed in the system design, LiDAR is used to measure distance in front of the cars. FRDM K64F uses I2C communication protocol to obtain the distance from LiDAR. Depending on the distance measured, FRDM K64F communicates with S32K144 using UART to send the appropriate motor control switch case. Depending on the switch case received by S32K144, motor speed is controlled using MBDT toolbox. As the distance between the vehicles decrease the speed of the motor reduces, thus adapting to the speed of the vehicle in front of it. If Lidar senses an obstacle that is very close to the vehicle, the vehicle reduces its speed and stops. At the same time, FRDM K64F sends a COAP message through KW41Z to the vehicle behind it using thread protocol. Once the vehicle behind receives this COAP message, it stops immediately.

Case 2: Autonomous surveillance using drones comprises of crash detection in vehicle-to-infrastructure communication. Crash between vehicles is detected using a combination of readings from Lidar, accelerometer in FRDM K64F and status of the COAP message. Once the crash has been detected, the

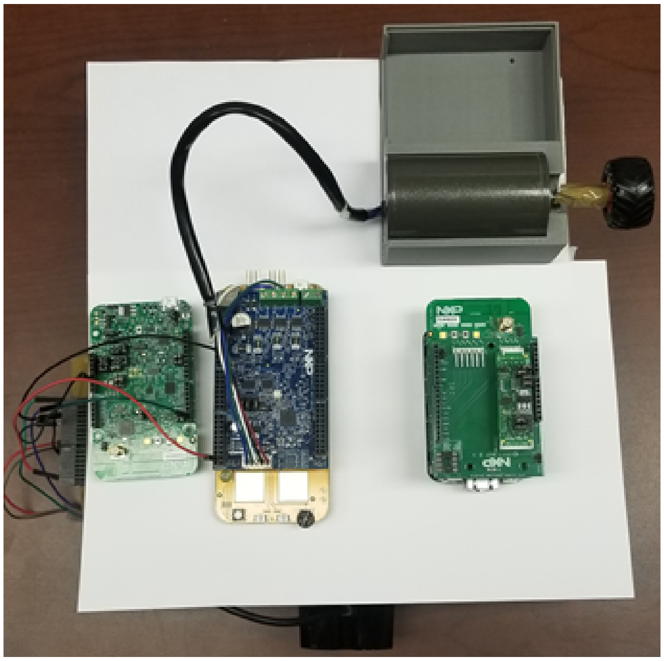

Fig. 6. Hardware Setup for One Vehicle

Fig. 7. Real Time Testing of V2V Communication using Thread with Antennas

GPS location of the vehicle is sent to the border routers placed on infrastructure such as road signals, street lights using thread protocol. The border router encrypts this location using ecc encryption and publishes it on the IBM cloud. The Raspberry Pi that is placed on the drone subscribes this updated GPS location and publishes it to the flight controller. This puts the drone in autonomous mode for aerial surveillance of the crash location.

XII. Conclusion

This paper demonstrates a new approach of using Thread as a wireless communication protocol, instead of Wi-Fi or Bluetooth, in V2V and V2I platforms and was tested in existing vehicles. The power consumption during wireless transmission of information is highly reduced by this approach. Also, the usage of UAV systems for surveillance during the time of crisis or emergency provides swift actions to accident locations due to less air traffic. This would help the rescue team to evaluate the magnitude of the accident and deploy resources in a short span of time. The future work of this system would be incorporating high quality, low power real-time cameras to the drone for better video quality and making the drones deliver medicines to highly congested traffic locations in the event of an accident. The challenges faced by the rescue teams in the event of any happenings could be highly reduced by this approach and could save many causalities.

XIII. Acknowledgement

We would like to show our gratitude to NXP for sharing their knowledge and feed-backs for this research.

References

[1] National Highway Traffic Safety Administration Vehicle to Vehicle Communication, , (Last Accessed: March 24, 2019). [Online] Available: https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication

[2] Silicon Lab Thread Fundamentals, (Last Accessed: April 10, 2019). [Online] Available: https://www.silabs.com/documents/public/user-guides/ug103-11-appdevfundamentals-thread.pdf

[3] Dronecode PX4 Autopilot User Guide, , (Last Accessed: May 5, 2019). [Online] Available: https://docs.px4.io/en/

[4] Infrastructure Communication, (Last Accessed: April 7, 2019). [Online] Available: https://www.automotivelectronics.com/vehicle-to-infrastructure-communication/

[5] Pixhawk Overview,(Last Accessed: March 2, 2019). [Online] Available: http://pixhawk.org/

[6] Electronics Projects Focus Wireless Sensor Networks and Applications, (Last Accessed: January 1, 2019). [Online] Available: https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/

[7] NXP's Model Based Design Toolbox, (Last Accessed: April 2, 2019). [Online] Available: https://www.nxp.com/support/developer-resources/run-time-software/automotive-software-and-tools/model-based-design-toolbox:MC_TOOLBOX

[8] Y. G. Hong. Problem statement of IoT integrated with edge computing, (Last Accessed: April 24, 2019). [Online] Available: https://tools.ietf.org/id/draft-hong-iot-edge-computing-01.html

[9] B. Stiller and C. Schmitt. DTLS-based Security with two-way Two-way authentication for IoT, (Last Accessed: April 24, 2019). [Online] Available: https://tools.ietf.org/html/draft-schmitt-two-way-authentication-for-iot-02

[10] NXP's A71ch Plug and Trust - The fast, easy way to deploy secure IoT connections hardware overview, (Last Accessed: April 24, 2019). [Online] Available: https://www.nxp.com/docs/en/application-note/AN12135.pdf

[11]] L. Zimmermann, N. Mars, M. Schappacher, and A. Sikora, Development of thread-compatible open source stack, Journal of Physics: Conference Series, vol. 870, no. 1, p. 012001, 2017, (Last Accessed: November 29, 2018). [Online]. Available: http://stacks.iop.org/1742-6596/870/i=1/a=012001

[12]] thread, Thread Network Architecture, (Last Accessed: November 29, 2018). [Online]. Available: https://www.threadgroup.org/Portals/0/documents/events/ThreadIntro.pdf