Journal of Scientific & Industrial Research
Vol. 83, October 2024, pp. 1095-1103
DOI: 10.56042/jsir.v83i10.7710

A Novel and Optimised Thread-based Virtual Traffic Light Framework

Parul Arora[1], Ritika Wason[1], Gagandeep Singh Narula[2]* & M N Hoda[1]
[1]Centre for Research and Innovation in Interdisciplinary Studies, Department of Computer Science and Applications
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), Delhi 110 063, India
[2]Guru Gobind Singh Indraprastha University (GGSIPU), Delhi 110 078, India

Received 01 January 2024; revised 25 June 2024; accepted 09 August 2024

Green intelligent transportation systems have gained tremendous popularity in the recent years. An important aspect of these systems is traffic congestion, which is rapidly becoming one of the most serious problems affecting urban areas. Also, these systems with conventional traffic lights or unregulated junctions result in wastage of time as well as high fuel consumption. Therefore, a new paradigm of thread-based Virtual Traffic Lights (VTL) utilizing a mesh network and Vehicular Ad hoc Networks (VANETs) is proposed in this manuscript. The framework is an apt solution to overcome real-time traffic problems because it is perfect, quick and simple to apply. The proposed framework uses minimal components and, most importantly, does not require any roadside infrastructure. With this new technology, the old system is being replaced by a smart digital system. Instead of putting traffic lights at every intersection, road side unit is being used as a road controller through which all the vehicles get an instant message telling them to drive through the intersection or wait for a moment instead of using traditional traffic lights. The simulation results indicate that the proposed scheme can increase the percentage of traffic flow and the average speed of vehicles with threaded VTL, thereby saving long hours of waiting time. Furthermore, the proposed threaded virtual traffic light framework reduces the average waiting time by 42.85% and increases the average vehicle speed of normal vehicles and emergency vehicles by 43.47% and 35.71% respectively as compared to conventional system, thereby reducing the traffic congestion. In addition to this, the proposed threaded VTL system is compared with existing recent studies and it is inferred that our approach yields the least average waiting time (3. 44 mins), highest average vehicle speed (60 Km/h) and the least computational time (5.26 mins) to execute the flow of traffic. The threaded protocol in the given study can be used to design smart and reliable homes by utilizing the features of optimization and scalability achieved by the proposed framework.

Keywords: Intelligent transportation systems, Mesh network, Thread-based virtual traffic light, Traffic congestion, VANETs

Introduction

According to United Nations (UN) estimates, over 1.3 million people primarily in the age group of 5–29 years succumb to road traffic fatalities each year.[1] Almost over 3000 people are killed in road accidents on a daily basis.[2] A graph representing the national loss incurred due to road accidents in total fatalities in the year 2022 is shown in Fig. 1. These grave numbers are not just an economic loss to the immediate family but also to the nation as a whole.[3] The United Nations General Assembly has thus set an ambitious target of reducing the number of deaths and injuries by road traffic crashes by half by 2030.[(1)] To achieve this goal, a technology driven driver assistance system is the need of the hour which can ensure safe travel.[4]

To achieve the above goal an intra-vehicular, vehicle-to-vehicle (V2V) communication enabled mechanism can be a befitting solution.[5] In the near future, autonomous vehicles shall constitute a considerable percentage of road traffic.[6,7] Hence, a V2V system can help in gathering and sharing information like vehicle speed, position of a vehicle with respect to other vehicles in the same location.[8] To translate this idea into reality a low-power Wireless Sensor Network (WSN) would be desirable for data transmission.[9] However, a single WSN may be susceptible to failures. Hence, we suggest that Vehicular Adhoc Networks (VANETs) can serve as the technology that will help vehicles change quickly if there are changes to traffic laws or how wireless networks are built and used.[10] For example, VANETs can be used to build and change wireless networks quickly and to introduce new traffic rules.[11] In VANETs, vehicles talk to each other using technology called Dedicated Short Range Communications (DSRC) and a frequency of 5.9 GHz.[12] By getting up-to-date information about traffic, Virtual Traffic Lights (VTLs) can help cut carbon emissions and improve energy use, as well as improve traffic and vehicle mobility.[13] This

*Author for Correspondence
E-mail: gagannarulaa09@gmail.com

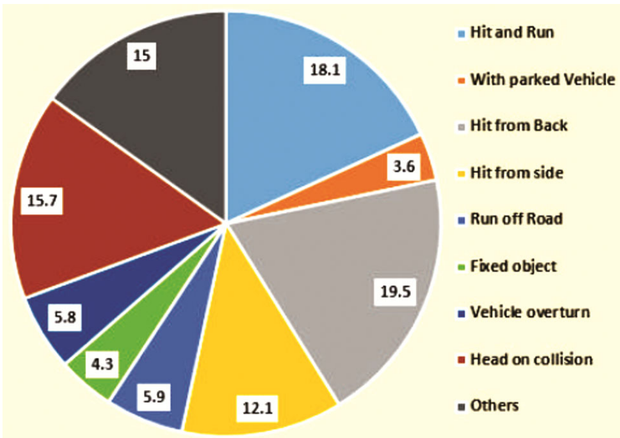

Fig. 1 — Distribution of types of collisions in total fatalities in 2022[(3)]

system can be fitted in vehicles with microcontrollers that can make use of new technologies.[14] The rest of this manuscript is organized as follows: Novelty of the study outlines how we are using the thread protocol for enabling a novel virtual traffic light system. Then, literature review and gap analysis gives a brief overview of existing intelligent light paradigms based on VANET. It is notable here that no thread-based systems are operational at the time of this writing. The next section discusses the major characteristics and the applications of the Thread protocol. In the Experimental setup section, we provide details about the hardware and software setup. The following part, working procedure, describes the stages involved in the implementation of our Nordic Semiconductor's nRF52840-DK. The following section, Implementation of Thread Network Protocol, provides screenshots of our thread protocol demonstration. The following part, performance evaluation, provides insight into outcomes on several performance measures. The next part discusses the study's potential uses. The following section addresses limitations. The final section discusses the conclusion and future scope of the manuscript.

Novelty of the Study

Together, a protocol is designed to ensure working of virtual traffic lights without centralized control system. A Virtual Traffic Light Leader (VTLL) is chosen from the different lane leaders at an intersection. Once a leader has been chosen, it is the job of the Individual Lane Leader (ILL) to send the signal to all of the cars in the same lane. The VTL method decides which lanes get the most attention. After letting everyone know that all lanes are clear, VTLL flashes its own green light.

In this study, an intelligent traffic management, which is one of the network applications of VANET[11], is shown off. Our strategy has led to shorter wait times for cars at intersections. With this new technology, the old system is being replaced by a smart digital system. Instead of putting traffic lights at every intersection, we use the RSU as a road controller. All vehicles get an instant message (through a Road Side Unit (RSU)) telling them to drive through the intersection or wait for a moment instead of using traditional traffic lights.

Literature Review and Gap Analysis

This section presents a few intelligent traffic light paradigms based on VANETs. Rzepecki *et al.*[15] proposed a roadside unit (RSU) using a signal control system for mitigating traffic congestion. This approach makes use of a vehicle priority scheduling algorithm to provide optimal driving speed. However, the method does not mention the complexities and time needed to form RSU. Liu *et al.*[16] developed a traffic signal control system for performing car-to-car communications. In this way, the incoming traffic signals are detected on the basis of location and traffic density. Moreover, this approach failed to validate the proposed framework using any simulation software. Tanwar *et al.*[17] used a multi-agent reinforcement learning signal control scheme to reduce the average waiting time and enhance the average speed of vehicles. Each intersection agent is used to share traffic information among neighboring agents. But this method failed to integrate multi-agents into a virtualized traffic environment. Tague *et al.*[18] proposed the installation of RSUs at each road intersection with the aim of performing vehicular communications. Despite this, the study did not provide any mechanism to deal with longer traffic jams created at intersections of RSU's. Vangimalla *et al.*[19] applied first-in-first out (FIFO) algorithm for scheduling of vehicles in virtualized traffic environment. This approach suffers from lock-out condition in which the vehicles with longer waiting time leads to congestion thereby ignoring lesser waiting time vehicles. Marksteiner *et al.*[20] proposed an embedded algorithm for reducing queuing delay at each of the RSU's. The proposed algorithm measures the length of lanes for the scheduling of vehicles. However, the approach failed to validate the algorithm in any simulation environment. Choudhary *et al.*[21] used a self-controlled traffic control scheme for monitoring several RSU intersections. It used infrared sensors for the detection of the number of vehicles and their average waiting time. But, the given approach led to higher waiting time due to multiple RSU's. Rajagopal *et al.*[22] used genetic algorithm for optimization of traffic signal management. The algorithm is supplied with velocity parameters based on the transmission area. It provides velocity recommendations based on traffic signals. Still, it suffers from the drawback that the optimization of traffic signals is not done properly as it cannot handle the arrival time of vehicles simultaneously. Mishra *et al.*[23] used a dictionary-based adaptive traffic signal algorithm. The method incorporates arithmetic coding, context modelling, and dictionary utilization. These dictionary based methods do not incorporate thread protocols and works like traditional traffic module. Khattak *et al.*[24] used thread network protocol for performing communications in smart cities thereby increasing the reliability and efficiency of thread protocol for upcoming projects. But, the study did not provide any mechanism for handling deadlock condition on roads in smart cities. Upadhyay *et al.*[25] utilized thread based mesh networking technology to transfer current traffic situations among the neighboring agents. However, the given approach failed to perform individual road side units with respect to current location. Naseri *et al.*[26] applied the usage of thread protocol technology in urban management cities. The study makes use of networking algorithms and priority scheduling to compute the shortest path in the network thereby ensuring smooth flow of traffic. But, the algorithms used in this study leads to higher computational time and resources.

Thread Protocol: Characteristics and Applications

Thread is an IP based wireless networking protocol designed for low-power connected products in home automation space.[27,28] The Thread technology operates under 2.4 GHz band and offers a secured mesh network for embedded, low power, low cost devices. The single hop range of a thread-based mesh network is limited to a few hundred meters only. The Thread Network can access Wi-Fi via router on 6LoWPAN protocol. Initially designed for home automation but this protocol is now used across several industrial applications as detailed in Table 1 below:

As indicated in Table 1 above, the thread protocol is the best solution for securing device to device communication in multiple modes without any single point of failure. Further unlike Zigbee, it is also open to application layer selection and allows systems to self-configure as well as operate in a scalable as well as reliable manner. The above advantages of the protocol have made it a natural choice for large commercial installations. Thus, after this detailed and minute analysis of the Thread protocol, we propose to apply the Thread to realize a real-time, personal virtual traffic light framework. The experimental

Table 1— Applications of thread Protocol

Purpose[Ref]	Description	Remarks
Drone monitoring using thread protocol in vehicular to vehicular network[5]	Integrates intra-vehicular communication with the Thread for adaptive cruise control in a vehicle during crisis.	Use of drones for aerial view detection in real-time, highly congested traffic locations would require robust testing.
Application of thread protocol in classification of arthritis using fog computing.[17]	A Thread and Bayesian network classifier based architecture to regularly monitor hand movements of the patients.	Data management can be a challenge in absence of an efficient cloud infrastructure.
Utilization of thread technology in mesh systems.[18]	Evaluates performance of Thread protocol for lighting systems in an office area.	Considerable performance of Thread protocol for wireless personal area networks under different traffic conditions was noted
Designing of a secured thread network.[19]	Vulnerabilities of Thread protocol that could impact the physical security of users have been evaluated.	Requires evaluation of devices or locators attached to thread.
Achieving interoperability in healthcare using thread protocol.[20]	To mitigate interoperability issues in remote locations	Thread Protocol with a proposed Over The Air (OTA) file transfer functionality, was successfully integrated into UAV design as a viable solution
Automation of homes using thread protocol.[21]	The low-power wireless protocols for interoperability have been compared in detail	A detailed analysis of the Thread has been presented indicating its suitability to home and building automation.
An IoT based thread protocol for wireless systems.[22]	Varied Wireless IoT protocols for a smart home scenario have been compared.	The protocol knows the limitations of network fluctuations.

details of our simulation are detailed in the following section.

Experimental Setup

We now elaborate the test environment's hardware and software setup. Certain equipment was put up in a standalone mode so that testing could be performed on four automobiles. Following hardware specifications are required as:

- Nordic Semiconductor nRF52840 Development Kit
- USB to microUSB cables
- Linux based computer with at least 3 USB ports
- Open battery nodes and open base nodes

Working Procedure

- ❖ Nordic Semiconductor's nRF52840- DK is a single-board development kit for the nRF52840 Multi-Protocol SoC, which supports Bluetooth Low Energy (BLE), 802.15.4 Thread, ANT/ANT+, and proprietary 2.4 GHz applications (System on Chip).
- ❖ Software used: - The nRF52840-DK can be utilized with the widely used Nordic toolchain software, including *Keil, IAR, and GCC*, according to the manufacturer. These tools allow for the programming and debugging of Segger J-Link OBs and other external target boards.
- ❖ The Development Kit includes four LEDs and buttons that may be programmed by the user, as well as edge connectors that provide access to all of the I/O and interfaces. NFC tags can be made functional with the use of an NFC antenna, which designers can incorporate in their projects.
- ❖ Construction of thread network: - After we have all of our terminal windows and screens in a place, we constructed our Thread network. Provide the FTD Commissioner with a default operational dataset. The operational dataset contains the settings for the Thread network.

Implementation of Thread Network Protocol

This CodeLab demonstrates how to use a minimal Open Thread programme to access the Open Thread administration and setup APIs via the command line interface (CLI). The goal of this exercise is to ping a virtual Thread device using a second virtual Thread device.

A basic thread network is depicted in Fig. 2(a). A fresh working data set is developed and marked as live. It then takes precedence over any previous working data sets. The operational information serves as a guide for the Thread network design.

As shown in Fig. 2(b), the dataset init new command is used to generate a new Operational Dataset populated with random values. It ought to serve as your principal data repository. After the IPv6 interface is up and running, the Thread protocol can begin. Wait a moment and check to see if your device has assumed the role of Thread Leader. The Leader is the node in the network responsible for giving out router IDs.

The IPv6 addresses assigned to Node 1's Thread interface are shown there in Fig. 2(c). Routing locators (RLOC's)[28] are dynamic and will change as the network architecture evolves and a Thread device changes statuses. These locators are assigned to our devices for enhancing scalability of the system and optimize incoming traffic. The endpoint identifier (EID) will remain unchanged regardless of any alterations made to the underlying network architecture.

Launch a bash shell for Node 2 in a new terminal inside the running Docker container. At this new bash prompt, you should enter the argument to begin a command line interface procedure. Thread Network Key and PAN ID are set up with the same information as Node 1's operational dataset. The second thread device emulation is shown in Fig. 2(d) which means that this dataset is currently being used.

Two simulated devices' ability to communicate across a Thread network was validated in our previous meeting. This is the only way IPv6 link-local traffic may be sent between devices. Thread boundary routers are required for nodes to connect with one another and the Internet. In this example, we'll stick with the tried-and-true two-node setup. In terms of authentication, the Commissioner will play the role of Thread Leader, while the Joiner will take on the role of Thread Router.

- ❖ While you are still at Node 1, you should initiate the process of appointing a Commissioner. A Joiner must be added to a commissioned Thread Network by a human administrator.
- ❖ Second, launch a second CLI process inside the Docker container's terminal. This is the second network node. It is necessary to activate the Joiner role on Node 2 using the J01NME Joiner Credential.

As shown in Fig. 2(e), the device (Node 2) has completed the verification process with the Commissioner (Node 1) and received the necessary

Thread Network credentials. If you want to make sure that Node 2 has successfully joined the network, you can do so by checking its current status by performing factory reset as shown in Fig. 2(f).

Recent applications of Thread Network Protocol

- It is a reliable and energy saving protocol which delivers faster response times and elevation in designing of smart homes.
- It mitigates the issues involved in designing Internet of things (IoT) based smart homes such as interoperability, security and reliability.

Performance Evaluation

Once a thread network has been implemented successfully, its performance can be evaluated using a number of criteria. The results of our simulations and tests show a significant reduction in average wait time, as well as improvements in efficiency, scalability, battery life, and latency. Threaded VTL performance is evaluated using the riverbed modeler AE 17.5 simulator.[27] The simulation area is around 1000 square meters in size and has four vertices. Microcontrollers with threading capabilities have been placed at all node intersections. Each vehicle has a broadcast range of 250 meters and employs IEEE 802.15.4 wireless technology connected via a mesh network. In the simulation, the number of vehicles and their speeds are both determined at random, with the total number of vehicles travelling anywhere from 200 to 800 kilometers per hour (km/h). We discuss our observations as shown in Table 2.

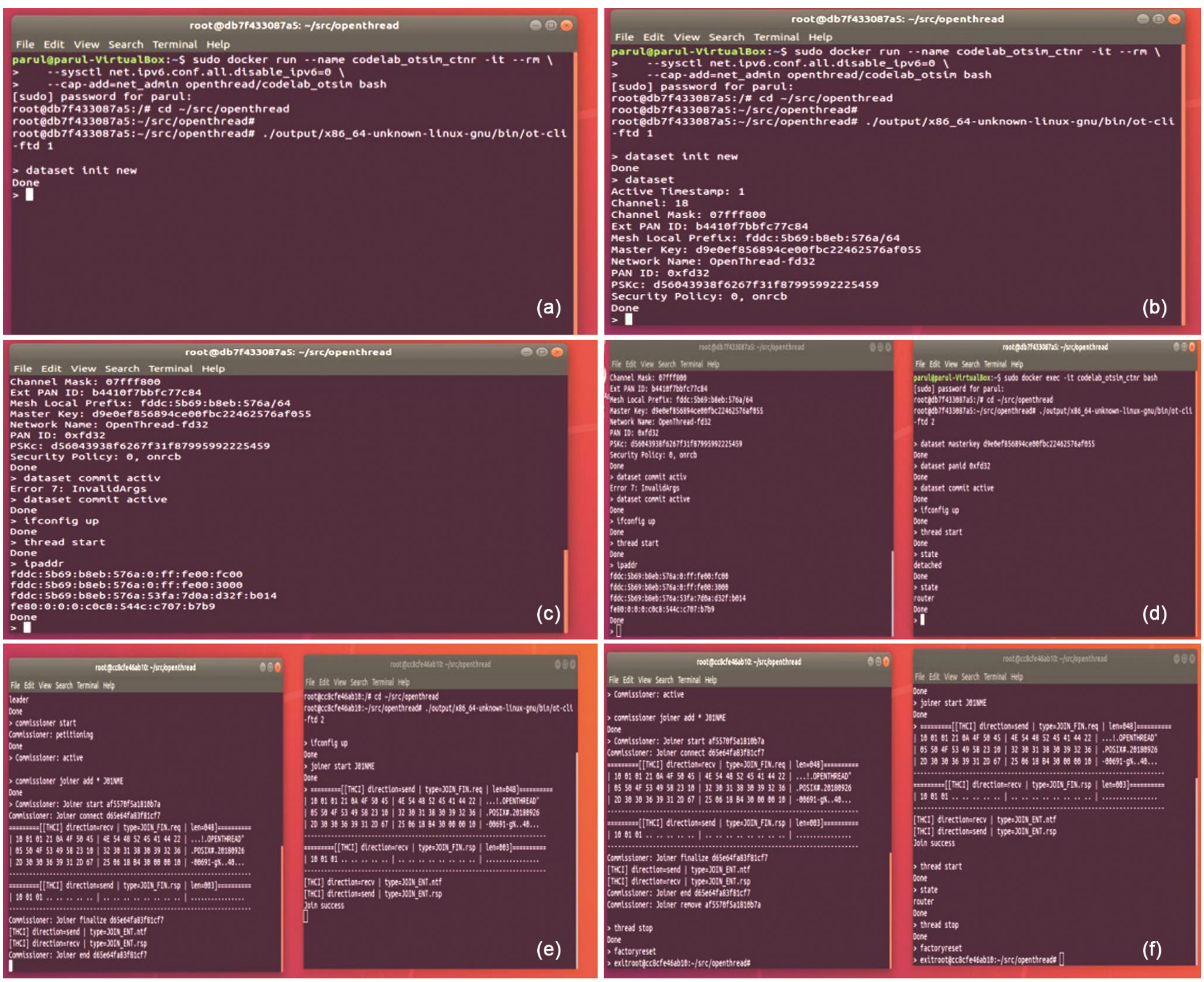

Fig. 2 — Thread network: (a) Establishment of basic thread network, (b) A new operational random dataset, (c) Initiating the thread network, (d) Starting the thread protocol in node-2, (e) Thread commissioning, and (f) Factory Reset of both nodes

Table 2 — Framework evaluation

Parameters	Observations	Depiction
Time spent waiting in a convoy of vehicles	Threaded VTL reduced the wait time on average by 42.857% when compared to regular VTL when the number of vehicles is high.	Fig. 3(a)
Large Traffic Concentrations	Threaded VTL increases average vehicle speed of normal vehicles by about 43.47%. Emergency vehicles could go at speeds like 30km/h as compared to 20km/h with traditional VTL (in case of 800 vehicles)	Fig. 3(b) Fig. 3(c)
Extent of battery life, speed of response, and density of available networks	Decreases energy expenditure, channel utilization, and communication delay in Sleepy End cars. Hence, electric or energy-harvesting vehicles will be more productive. Threaded VTL uses an optimization to increase productivity and speed.	Fig. 4

Results & Discussions

This section visualizes the results as discussed in Table 2 above.

(a) Time Spent Waiting in a Convoy of Vehicles

As visible from Fig. 3(a), threaded VTL helped achieved reduced waiting for vehicles especially in cases of higher vehicle density. The waiting time (mins) for vehicles is directly proportional in case of conventional VTL approach while it is indirectly proportional to the proposed threaded VTL approach. As the number of vehicles rises exponentially, threaded VTL approach reduces the average waiting time thereby mitigating traffic congestion and ensuring smooth flow of traffic. Initially, the number of vehicles is 200 and the waiting times are 2 minutes (VTL) and 1 minute (threaded VTL) respectively. When the number of vehicles become 400, the waiting times are 4 minutes (VTL) and 2 minutes (threaded VTL) respectively. Similarly, in case of 600 vehicles, the waiting times are 6 minutes (VTL) and 4 minutes (threaded VTL) respectively. Lastly, in case of 800 vehicles, the waiting times are 9 minutes (VTL) and 5 minutes (threaded VTL) respectively. On computing average waiting time in both the approaches, an inference is derived that the threaded VTL approach in overall reduces the average waiting time of vehicles by 42.857%.

(b) Large Traffic Concentrations

Threaded VTL also demonstrated improved performance in terms of speed improvement for normal as well as emergency vehicles for higher vehicle concentration road patches. This is depicted for normal vehicles in Fig. 3(b) and for emergency vehicles in Fig. 3(c).

For normal vehicles, the average vehicle speed (Km/h) is inversely proportional to the number of vehicles in case of conventional VTL approach while it is directly proportional to the proposed threaded VTL approach. As the number of vehicles rises exponentially, threaded VTL approach increases the average speed of vehicles thereby ensuring smooth flow of traffic. Initially, the number of vehicles is 200 and the average speeds are 40 Km/h (VTL) and 60 Km/h (threaded VTL) respectively. When the number of vehicles become 400, the speeds become 35 Km/h (VTL) and 45 Km/h (threaded VTL) respectively. Similarly, in case of 600 vehicles, the speeds become 25 Km/h (VTL) and 40 Km/h (threaded VTL) respectively. Lastly, in case of 800 vehicles, the values become 15 Km/h (VTL) and 20 Km/h (threaded VTL) respectively. On computing average vehicle speeds in both the approaches, an inference is derived that the threaded VTL approach increases the average speed of normal vehicles by 43.47%.

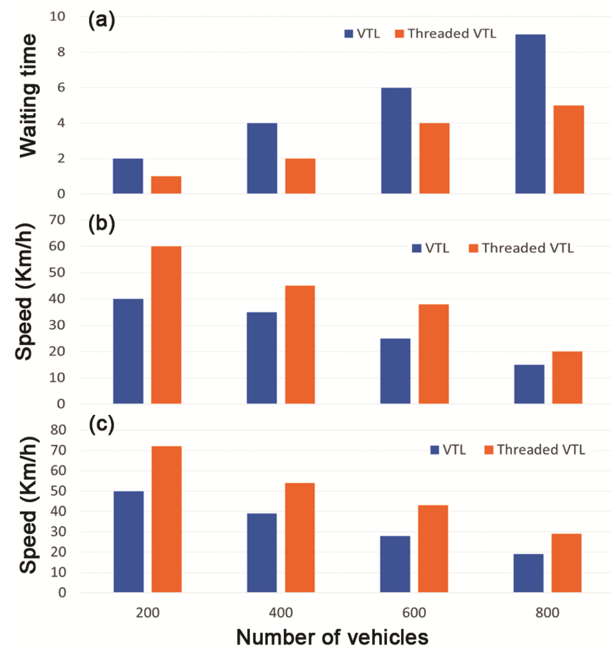

Fig. 3 — Dependency of different parameters on number of vehicles(a) Number of vehicles vs. Waiting time, (b) Number of vehicles vs. Average speed of normal vehicles, and (c) Number of vehicles vs. Average speeds of emergency vehicles

For emergency vehicles as shown in Fig. 3(c), the average vehicle speed (Km/h) is inversely proportional to the number of vehicles in case of conventional VTL approach while it is directly proportional to the proposed threaded VTL approach. As the number of emergency vehicles rises exponentially, threaded VTL approach increases the average speed of vehicles thereby ensuring smooth flow of traffic. Initially, the number of vehicles is 200 and the average speeds are 50 Km/h (VTL) and 70 Km/h (threaded VTL) respectively. When the number of vehicles become 400, the speeds become 40 Km/h (VTL) and 50 Km/h (threaded VTL) respectively. Similarly, in case of 600 vehicles, the speeds become 30 Km/h (VTL) and 40 Km/h (threaded VTL) respectively. Lastly, in case of 800 vehicles, the values become 20 Km/h (VTL) and 30 Km/h (threaded VTL) respectively. On computing average vehicle speeds in both the approaches, an inference is derived that the threaded VTL approach increases the average speed of emergency vehicles by 35.71%

In addition to this, we collate our proposed approach results with the existing recent studies.[19,20,21,24,25,26] in terms of average waiting time (mins), average vehicle speed (Km/h) and computational time (CT) (mins). The results are shown in Table 3 below.

It can be seen that average waiting time (mins) of recent studies and the proposed approach is 8.27, 7.23, 9.19, 6.25, 10.17, 9.55 and 3.44 respectively. In both cases, it is the lowest for the proposed approach with the highest average vehicle speed (Km/h) as compared to recent studies. It implies that the proposed approach is able to mitigate deadlock or lock-out condition. The average vehicle speed (Km/h) for the proposed approach is the highest (60 Km/h) as compared to the recent studies. It is also observed that it takes less time to maintain traffic flow in the proposed system (5.26 mins) as compared to recent studies (15.67 mins, 13.12 mins, 11.19 mins, 15.23 mins, 9.12 mins and 10.55 mins) which indicates better performance of our approach.

Table 3 — Comparative analysis of the proposed approach with existing recent studies (overall)

Approaches / Algorithms	Avg. waiting time (mins)	Avg. vehicle speed (Km/h)	CT (mins)
Vangimalla *et al.*[19]	8.27	40	15.67
Marksteiner *et al.*[20]	7.23	35	13.12
Choudhary *et al.*[21]	9.19	45	11.19
Khattak *et al.*[24]	6.25	15	15.23
Upadhyay *et al.*[25]	10.17	25	9.12
Naseri *et al.*[26]	9.55	20	10.55
Proposed (threaded VTL)	3.44	60	5.26

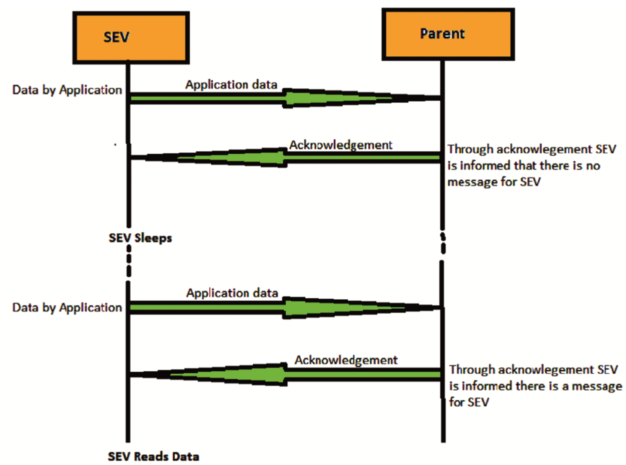

Fig. 4 — Enhanced framing function in threaded VTL

(c) Effect of SEV on using Improved Framing

The increased scalability of Thread Networks allows for a greater number of Thread Devices to coexist in a given area. In order for a SEV to know if its parent has data, it must transmit a Data Request command message, as required by IEEE 802.15.4. As a result, there will be need of more people using the internet and more people communicating specifically with each other. When a SEV sends information to its parent via EFP, the parent will confirm receipt of the information. This SEV operation has no overhead, thus no extra octets are transmitted. How the SEV can use improved framing to choose between resting and reading is shown in Fig. 4.

These advancements pave the way for the widespread adoption of low-power wireless vehicles in existing infrastructure networks. Larger bandwidth network segments, such as Ethernet or Wi-Fi, can connect many thread networks or partitions, and vehicles at the Thread Network's edge that helps in regulating Internet Protocol (IP) forwarding across subnet boundaries in a consistent fashion.

Conclusions

This study presents a novel method for creating a thread-based virtual traffic indicator that can quickly adapt to the current road and traffic conditions by utilizing upcoming vehicular communication technologies. A protocol is proposed that enables the construction of virtual traffic lights without requiring a centralized control system. The simulation results indicate the validation of the proposed framework in terms of reduced average waiting time and higher average speed of vehicles thereby creating a smooth flow of traffic. This work holds immense potential in reducing the number of deadlocks among vehicles by creating optimized traffic flow with lesser number of accidents. However, its limitation lies in the need of addressing the effect of non-line-of-sight for vehicular communications. As future scope, the work can be extended to incorporate some of the features such as sending dynamic information to pedestrians related to road traffic, developing routing algorithm and utilization of priority lanes for vehicles.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

1 Naik D, Bauri A & De T, Renewable Energy Aware Traffic Grooming in Hybrid Optical and WiMAX Networks, *Wireless Pers Commun*, **123** (2022) 2317–2339, https://doi.org/10.1007/s11277-021-09243-z.

2 Sankaranarayanan M & Mathew S, Pre-processing framework with virtual mono-layer sequence of boxes for video based vehicle detection applications, *Multimed Tools Appl*, **80** (2021) 1095–1122, https://doi.org/10.1007/s11042-020-09587-x.

3 Oladimeji D, Gupta K, Kose N A, Gundogan K, Ge L & Liang F, Smart transportation : An overview of technologies, *Sensors*, **23** (2023) 1–32.

4 Thiyagarajan S, Kumar K & Renuka S, Performance estimation of joint user cluster pairing for 2-SUs CR-multi-user NOMA downlinks system, *Int J Inf Technol*, (2023), https://doi.org/10.1007/s41870-023-01672-2.

5 Chitanvis R, Ravi N, Zantye T & El-Sharkawy M, Collision avoidance and drone surveillance using thread protocol in V2V and V2I communications, *Proc Natl Conf on Aerospace Electronics* (IEEE), 2019, 406–411.

6 Olaverri-Monreal C, Errea-Moreno J & Díaz-Álvarez A, Implementation and evaluation of a traffic light assistance system based on V2I communication in a simulation framework, *J Adv Transp*, (2018) 378597, https://doi.org/10.1155/2018/3785957.

7 Taeihagh A & Lim H S M, Governing autonomous vehicles: Emerging responses for safety, liability, privacy, cybersecurity, and industry risks, *Transp Rev*, **39** (2019) 103–128, https://doi.org/10.1080/01441647.2018.1494640.

8 Tomar I, Sreedevi I & Pandey N, State-of-art review of traffic light synchronization for intelligent vehicles: Current status, challenges, and emerging trends, *Electronics*, **11(3)** (2022) 465, https://doi.org/10.3390/electronics11030465.

9 Reyes-muñoz A & Guerrero-ibáñez J, Vulnerable road users and connected autonomous vehicles interaction: A survey, *Sensors*, **22(12)** (2022) 4614, https://doi.org/10.3390/s22124614.

10 Perveen A, Abozariba R & Patwary M, Dynamic traffic forecasting and fuzzy-based optimized admission control in federated 5G-open RAN networks, *Neural Comput Appl*, **35** (2023) 23841–23859, https://doi.org/10.1007/s00521-021-06206-0.

11 Ritika W, Vishal J, Narula G S, Anupam B & Mandeep K, Smart robotics for smart healthcare, *Adv Rob Mech Eng*, **1(5)** (2019) 73–74, doi: 10.32474/ARME.2019.01.000121.

12 Wu Q, Wu J, Shen J, Yong B & Zhou Q, An edge based multi-agent auto communication method for traffic light control, *Sensors*, **20(15)** (2020) 4291, https://doi.org/10.3390/s20154291.

13 Damadam S, Zourbakhsh M, Javidan R & Faroughi A, An intelligent IoT based traffic light management system: Deep reinforcement learning, *Smart Cities*, **5(4)**(2022) 1293–1311, https://doi.org/10.3390/smartcities5040066.

14 Thomas E, McCrudden C, Wharton Z & Behera A, Perception of autonomous vehicles by the modern society: A survey, *IET Intell Transp Syst*, **14** (2020) 1228–1239, https://doi.org/10.1049/iet-its.2019.0703.

15 Rzepecki W & Ryba P, IoTSP: Thread mesh vs other widely used wireless protocols - Comparison and use cases study, *Proc Int Conf on Internet Things Cloud* (IEEE), 2019, 291–295, doi: 10.1109/FiCloud.2019.00048.

16 Liu Y, Pang Z & Dan G, Latency analysis of wireless networks for proximity services in smart home and building automation: The case of thread, *IEEE Trans Ind Electron*, **14(7)** (2019) 4856–4867, doi: 10.1109/ACCESS.2018.2888939.

17 Tanwar S, Vora J, Kaneriya S, Tyagi S, Kumar N, Sharma V & You I, Human arthritis analysis in fog computing environment using bayesian network classifier and thread protocol, *IEEE Consum Electron Mag*, **9(1)** (2020) 88–94, doi: 10.1109/MCE.2019.2941456.

18 Akestoridis D G, Sekar V & Tague P, On the security of thread networks: Experimentation with openthread-enabled devices, *Proc Int Conf Secur Priv Wirel Mob Networks* (ACM), 2022, 233–244, https://doi.org/10.1145/3507657.3528544.

19 Vangimalla S R & El-Sharkawy M, Interoperability enhancement in health care at remote locations using thread protocol in UAVs, *Proc Int Conf Electronics* (IEEE), 2018, 2821–2826, doi: 10.1109/IECON.2018.8592759.

20 Marksteiner S, Jimenez V J, Valiant H & Zeiner H, An overview of wireless IoT protocol security in the smart home domain, *Proc Int Conf on Internet Things - Bus Model, Users, Networks* (IEEE) 2017, 1–8, doi: 10.1109/CTTE.2017.8260940.

21 Choudhary P & Dwivedi R K, A novel algorithm for traffic control using thread based virtual traffic light, *Int J Inf Tecnol*, **14** (2022) 115–124, https://doi.org/10.1007/s41870-021-00808-6.

22 Rajagopal B G, Intelligent traffic analysis system for Indian road conditions, *Int J Inf Tecnol*, **14** (2022) 1733–1745, https://doi.org/10.1007/s41870-020-00447-3.

23 Mishra S, Maheshwari R & Grover J, Investigating the performance of a vehicular communication system based on visible light communication (VLC), *Int J Inf Tecnol*, **14** (2022) 877–885, https://doi.org/10.1007/s41870-021-00834-4.

24 Khattak S B A, Nasralla M M, Farman H & Choudhury N, Performance evaluation of an IEEE 802.15.4-based thread network for efficient internet of things communications in smart cities, *Appl Sci*, **13** (2023) 7745, https://doi.org/10.3390/app13137745.

25 Upadhyay P, Sharma S & Guangdong B, Symbolic verification of mesh commissioning protocol of thread, *Proc Int Conf on Innov Sof Eng* (ACM), 2024, 1–4, https://doi.org/10.1145/3641399.3641446.

26 Naseri A, Estelaji F, Samani A, Omidifar R, Zahedi R & Yousefi H, Simulating the performance of HOV lanes for optimal urban traffic management, *Transp Res Interdisc Persp*, **23** (2024) 101010, https://doi.org/10.1016/j.trip.2023.101010.

27 Tabibi Z, Schwebel D C & Zolfaghari H, Road-crossing behavior in complex traffic situations: A comparison of children with and without ADHD, *Child Psych Hum Dev*, **53** (2022) 1186–1193, https://doi.org/10.1007/s10578-021-01200-y.

28 Mazumdar S, Scionti A & Zuckerman S, NoC-based hardware software co-design framework for dataflow thread management, *J Supercomput*, **79** (2023) 17983–18020, https://doi.org/10.1007/s11227-023-05335-8.