Minimizing Response Time of IoT-Based Traffic Information System Through A Decentralized Server System

Maryam Hassanlou
California State University, San Marcos, San Marcos, California, USA

Yi Sun
California State University, San Marcos, San Marcos, California, USA

Fang Fang*
California State University, San Marcos, San Marcos, California, USA

Many metropolises seek to relieve traffic congestions and reduce vehicle accidents by implementing intelligent traffic information systems. These systems manage continuous communication between vehicles, various roadside Internet of Things (IoT) devices and central servers in real time for traffic control and vehicle guidance and navigations. Short response time is critical to the success of these time-sensitive systems. For a small area, a system with centralized server architecture may just work fine. For a larger area with more IoT devices and traffic to manage, however, the system may experience excessive response time as a result of increased network distance and constrained server processing capacity. In this paper, we propose a decentralized server system to properly manage and reduce service response time. We also develop a binary nonlinear constrained programming model, and numerical results are provided using the MATLAB MINLP solver to support the proposed model.

* Corresponding Author. E-mail address: fangfang@csusm.edu

I. INTRODUCTION

Many metropolises seek to reduce traffic congestions, vehicle accidents, and pollutions by implementing intelligent traffic information systems (Allström et al., 2016). These systems manage continuous communication between vehicles, traffic control systems, and various roadside Internet of Things (IoT) devices with sensors, and processing servers. They measure real-time traffic density and weather conditions and control the traffic congestion on the road through dynamic management of traffic signals.

In today's world, vehicles are increasingly becoming connected and ready to interact with nearby connected vehicles and IoT safety devices along the roadside by sending and receiving data continuously. In particular, the development and deployment of autonomous vehicle can benefit greatly of such connections (Lu et al., 2019). Such an infrastructure is both supported by private industry and by government agencies as well (https://www.its.dot.gov/cv_basics/index.htm). Data collected by these IoT devices is then fed to a central server in real time, which, in turn, analyzes and gives instructions back to the IoT devices. The IoT devices will then relay back to the traffic control systems and/or the vehicles directly to help with traffic controls in order to avoid accidents and shorten travel times.

The term Internet of Things (IoT) was coined by Kevin Ashton of Procter & Gamble in 1999 (Kevin, 2009). IoT has since then received significant attention both in academia and industry during the past decade. It prescribes a world where numerous smart objects are connected to each other with no human intervention. IoT has been used in many smart applications for healthcare, home and office, agriculture, equity trading (Sun and Hassanlou, 2019), etc. In transportation, various IoT sensors are available and many are currently deployed to help control and manage the traffic information systems efficiently.

In general, an intelligent traffic information system needs to offer fast services to keep up with fluid, sometimes chaotic, and continuous traffic. The success of these time-sensitive systems is partially determined by their service response time. For a small area, a centralized server architecture with one central server and multiple IoT devices may work just fine. For a larger area with more IoT devices and a high volume of traffic, however, the system may experience excessive response time as a result of increased network distance and constrained server processing capacity. Properly managing and reducing response time is a critical requirement in order for traffic information systems to achieve their goals.

An alternative solution is to deploy a decentralized traffic information system. In such a system, there can be three major players: a central server, multiple local servers, and numerous IoT devices. Vehicles communicate directly with IoT devices nearby in real time, report key vital statistics, including speed, vehicle type, and destination, and request services for traffic guidance. IoT devices then relay this information directly to local servers nearby for speedy processing. Local servers, subsequently, process the information and give guidance back to the vehicles through the IoT devices they interact with. At the same time, the local servers also serve as intermediaries between IoT devices and the central server. The local servers forward important local traffic information, gathered by IoT devices in the field, to the central server. The central server, in turn, processes the information at an aggregated level and communicate back to local server for global traffic directions. In essence, the central server is responsible for managing all of the communications with IoT devices through intermediary local servers and overall traffics in the entire metropolitan area (figure 1).

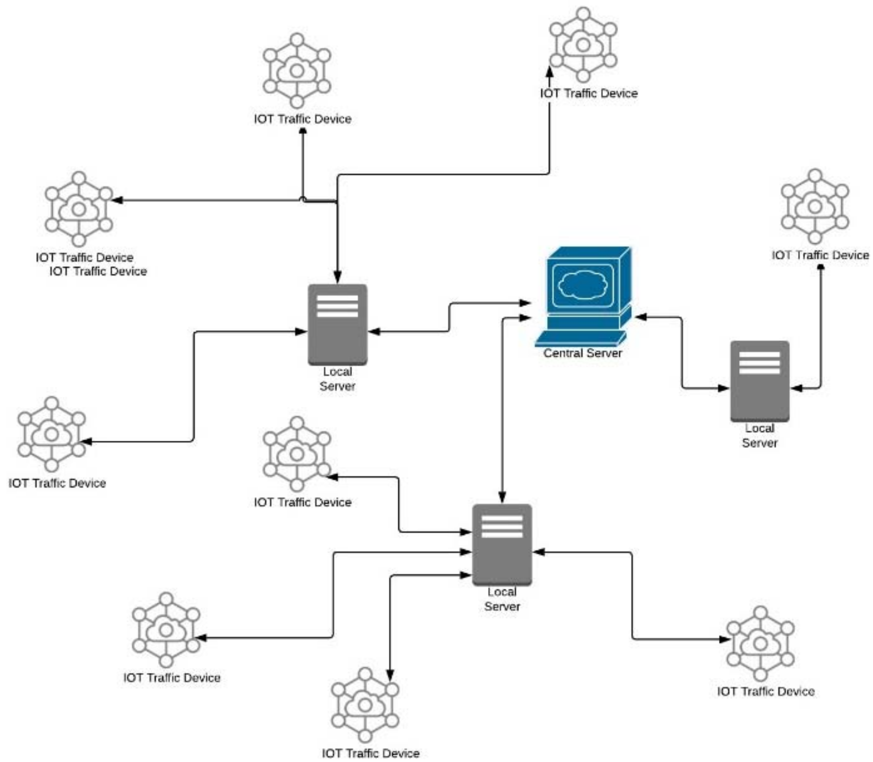

FIGURE 1. DECENTRALIZED TRAFFIC INFORMATION SYSTEM

The usefulness of an intelligent traffic information system depends on accurate and fast communication between vehicles and servers. Speedy network and fast processing servers make the entire system responsive to the requests from vehicles. As a result, the performance of such time-sensitive decentralized service is largely measured by its response time. Response time includes local processing time and network response time. Network response time is largely determined by network latency. Network latency refers to the amount of time that a packet of data takes to travel from one location to another on a network (Johansson, 2000). Minimizing service response time, as a result, requires reducing local processing time and decreasing the network latencies between servers and clients. Network latency is closely related to the physical proximity between IoT devices and their assigned local servers. Instead of connecting IoT devices to a distant central server, we can locate many local servers physically near them for service request to reduce overall network latency. A local server handles much of the request of IoT devices in real time and only needs to connect with the central server for global traffic management.

The strategic placement of the local servers on a network, therefore, becomes critical in reducing network latency and service response time. Since there will be many communications between local servers and the central server for global traffic management, the distance between them will also need to be reduced by optimal locating the central server on the same network. To decrease local server processing time, we can choose more capable server equipment and software package within a budget.

The main purpose of this research is to provide a framework that can guide a metropolitan area to locate and manage its local and central servers to improve traffic services. We developed a binary nonlinear constrained programming model with budget and service response time constraints. We then solved the problem with MATLAB R2019a Mixed Integer Nonlinear Programming (MINLP) solver. Sensitivity analysis was also conducted to provide additional managerial insights.

II. LITERATURE REVIEW AND RATIONALS

IoT devices are widely used in smart cities, particularly for managing traffics (An et al., 2011, Nemtanu et al., 2014, Ramazani and Vahdat-Nejad, 2014, Sukode et al., 2015). Allström et al. (2016) suggested that to manage and control traffic flows, the IoT devices need to capture the conditions of the road traffic with speed, flow, and density on a specific segment of the road. They proposed a framework to utilize the various traffic management sources efficiently in the context of traffic management and analyzed how different types of traffic models and algorithms can use the data sources and key functionalities of active traffic management such as short-term prediction and control. Rath (2018) argued that the growth of population and vehicles causes traveling delays and contributes to environmental pollution and therefore recommended a smart IoT-based system to alleviate the problem. Al-Sakran (2015) proposed an intelligent traffic administration system, based on IoT, which features low cost, high scalability, high compatibility, and is easy to upgrade in order to replace traditional traffic management system to improve road traffic tremendously.

Avasalcai et al. (2018) suggested that for real-time applications with fast response times requirement, fog (Flavio and Milito, 2012) and edge computing (Shi and Dustdar, 2016) will be the key infrastructures for deployment. Both methods locate computing resources closer to IoT devices. Raptis et al. (2018) argued that the distribution of data generated by IoT technologies needs to be improved continuously. A centralized system with data being transferred back and forth in the network may lead to severely sub-optimal paths and communication overhead and ultimately increase overall network latency. To solve the problem, they proposed an edge data distribution system where services are distributed to nodes near IoT devices.

In particular, for an IoT-Based Traffic Information System to work efficiently, network latency needs to be carefully managed and reduced if possible. Traffic IoT sensors are implemented on a distributed network. Service requests from IoT devices generate many messages to discover, negotiate, and invoke these services for traffic management. In addition to technical consideration, managerial issues are also important factors to the success of system. All cities face budget and procurement constraints and need to work with them judiciously. In this study, our model will minimize overall response time by optimally locating local/intermediary servers and a central server with budget constraint and maximum response time constraint to serve all IoT devices connected on the network.

Currently, the study of the connections of IoT devices, vehicles and traffic information system as a technology solution has received much attention as indicated above. However, as an operation management issue, in particularly, as a server facility location problem, the study is still in its infancy. Nevertheless, given the importance of network latency and critical customer service response time, especially in a large metropolitan area, we provide a novel multi-level and capacitated binary nonlinear programming model to help address the problem.

III. DECISION MODEL

For convenience, we assume there is a network where we can locate IoT devices, local servers, and one central server. We assume that J number of IoT devices have already been deployed and each will generate a demand for service D_j. Given the fluidity of the traffic condition, we assume D_j is stochastic. There will be one central server and M different type of local servers we can purchase at price P_m with service capacity CP_m. We assume servers with higher capacity will command a higher price. On the same network, there are I possible locations for local servers and K possible locations for the central server location. The distance between local server I and center server k is f_{ik} and the distance between local server location i and IoT device j is d_{ij}. The fixed cost of locating a local server on location i is FI_i and the fixed cost of locating the central server on location k is FS_k. We further assume that the maximum tolerable response time for service is T and the total budget is B.

TABLE1. SUMMARY OF NOTATION

Parameters
M: number of local server types; m=1 … M
J: total number of IoT devices; j= 1… J
I: possible locations for local servers; i= 1…I
K: possible locations for central server location; k= 1…K
D_j: demand from each IoT device j (stochastic)
CP_m: Capacity (total number of demands that can be serviced) of local server type m
d_{ij}: distance between local server location i and IoT device j
f_{ik}: distance between local server location i and central server location k
FI_i: fixed cost of locating a local server on location i
FS_k: fixed cost of locating central server on location k; ($FI_i < FS_k$)
P_m: price of local server type m
P_c: price of central server
t: time to receive data per unit of distance
T: maximum tolerable response time (if the response time exceeds T, it leads to time out)
B: total available budget
Decision Variables
X_{mi}: binary variable; takes 1 if local server m is located on location i
Y_k: binary variable; takes 1 if central server is located on location k
Z_{ij}: binary variable; takes 1 if IoT j gets service from local server located on i_{th} location

1. First, the deterministic version of the model is formulated as:

$$Min\ \text{P1}= \sum_{j=1}^{J}\sum_{i=1}^{I} d_{ij}Z_{ij} + \sum_{i=1}^{I}\sum_{k=1}^{I} f_{ik}Y_kX_{mi}$$

s.t.:

$$X_{mi} - Z_{ij} \geq 0 \quad ; \quad \forall i \qquad (1)$$

$$\sum_{j=1}^{J} D_jZ_{ij} \leq CP_mX_{mi} \quad ; \quad \forall i \,\&\, m \qquad (2)$$

$$\sum_{i=1}^{I} Z_{ij} = 1 \quad ; \quad \forall j \qquad (3)$$

$$\sum_{m=1}^{M} X_{mi} \leq 1 \quad ; \quad \forall i \qquad (4)$$

$$\sum_{k=1}^{K} Y_k = 1 \quad ; \quad \forall k \qquad (5)$$

$$t.d_{ij}Z_{ij} \leq T \quad ; \quad \forall i,j \qquad (6)$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} P_mX_{mi} + \sum_{i=1}^{I}\sum_{m=1}^{M} FI_iX_{mi} + \sum_{k=1}^{I} FS_kY_k + \text{Pc} \leq B \qquad (7)$$

$$X_{mi}, Z_{ij}, Y_k \in \{0,1\}, \ \forall m,i,j,k$$

The objective function P1 is designed to minimize the overall response time of the entire system. As the response time is a linear function of distance so P1 is formulated to minimize the total traveled distance for the required communication between IoT devices and local servers and between local servers and the central server. The proposed model is to optimize the location allocation problem subject to seven constraints.

Constraint (1) is to ensure that we only assign IoT device j to location i if a local server is to be deployed on location i.

Constraint (2) is to ensure that total IoT demand for connecting to each local server doesn't exceed the local server's capacity. Capacity is determined by the types of local server deployed.

Constraint (3) is to ensure that an IoT device should be assigned to one local server to respond the demand.

Constraint (4) is to ensure that on each possible location, a maximum of one local server can be located.

Constraint (5) is to ensure that only one central server should be located.

Constraint (6) is designed for preventing time out in service. It helps to ensure minimum service levels and the service time doesn't exceed the maximum tolerable time.

Constraint (7) is formulated for satisfying the budget limitation.

IV. CHANCE CONSTRAINED PROGRAMMING

In CCP, the objective function should be achieved with the stochastic constraints held at least α of time, where α is provided as an appropriate safety margin by the decision maker (Hassanlou, 2016).

Assume that x is a decision vector, ξ is a stochastic vector, and gj(x, ξ) are stochastic constraint functions, j= 1, 2… p. Since the stochastic constraints gj(x, ξ) ≤ 0, j= 1, 2… p does not define a deterministic feasible set, they need to be held with a confidence level α. Thus chance constraint is represented as follows (Liu, 2009):

$$\Pr\{ g_j(x, \xi) \le 0,\ j= 1, 2, \dots, p \} \ge \alpha \quad (8)$$

Which is considered the same α for all stochastic constraints, and when we want to assume that they are different, it can be shown as follows:

$$\Pr\{ g_j(x, \xi) \le 0\} \ge \alpha_j ,\ j= 1, 2, \dots, p \quad (9)$$

Theorem (1): Assume that the stochastic vector $\zeta = (a_1, a_2, \dots, a_n, b)$ and the function g(x, ξ) has the form $g(x, \xi) = a_1x_1 + a_2x_2 + \cdots + a_nx_n - b$. If a_i and b are assumed to be independently normally distributed random variables, then Pr { g(x, ξ) ≤ 0} ≥ α if and only if

$$\sum_{i=1}^{n} E[a_i]x_i + \Phi^{-1}(\alpha)\sqrt{\sum_{i=1}^{n} Var[a_i]x_i^2 + V[b]} \le E[b] \quad (10)$$

Where Φ is the standardized normal distribution function. The proof of the above theorem is in Liu (2009).

In this paper, we assume Dj, potential demand from each IoT device j, is stochastic and it follows normal distribution so its notation will be changed to a random variable as $\widetilde{D_j}$. In the proposed model, constraint (2) is the only constraint that includes stochastic parameter $\widetilde{D_j}$ so using equation (10), it is turned to chance constraint as following:

$$\sum_{j=1}^{J} E[\widetilde{D_j}]Z_{ij} + \sum_{j=1}^{J} \Phi^{-1}(\alpha)\sqrt{Var[\widetilde{D_j}]Z_{ij}{}^2} - CP_mX_{mi} \le 0; \quad \forall\, i, m \quad (11)$$

This proposed model is a Binary Constrained NLP where it includes one nonlinear constraint and objective as well. The model includes a collection of constraints: equality, inequality, linear and nonlinear constraints. This is a NP-hard combinatorial optimization problem. We employed MATLAB R2019a Mixed Integer

Nonlinear Programming solver to solve the problem. Then, we conducted numerical experiments to demonstrate the validity and robustness of the model. Using adequate sensitivity analysis, we evaluated the agility of model in responding to the changes on parameters to explore the behavior of model in extreme values.

V. NUMERICAL EXAMPLE

The following hypothetical numerical example along with the sensitivity study in the next section were selected to demonstrate the agility and robustness of the proposed model.

For the experiment, we created a network of communication to locate IoT devices, local servers and central servers. The locations of IoT devices were known. Each of the IoT devices generated multiple requests per unit of time with a normal distribution with a mean of 100 and standard distribution of 20 in order to address the demand uncertainty. As mentioned in the previous section, we deployed Chance Constrained Programming to handle uncertain parameters embedded in the model, so we needed to set the confidence level (α) to reflect the level of satisfaction for chance constraint (11). In this example we set the α as 0.9.

We limited the number of deployed IoT devices to 200. There were 20 possible locations for the local servers. Each location could host one of the three different types of local servers. The three different type of local servers cost $10,000, $20,000 and $30,000 respectively. They also offered different capabilities of handling 10,000, 30,000, and 50,000 requests respectively. The fixed costs of locating local servers on available locations were randomly generated and ranged between $1,000 and $5,000.

There were 10 possible locations for the central server. The fixed cost of deploying a central server at an available location was randomly generated in a range between $10,000 and $50,000. The price of the central server was set at $100,000.

The distance between the local server location and IoT device and the distance between local server location and central server location were all randomly generated and ranged between 100 and 5000 feet.

The time to transmit data per mile was assumed to be 8.2 microseconds (Sherman, 2019). The maximum tolerable response time was set at 3 microseconds. The overall budget was $300,000.

We ran the data using MATLAB MINLP solver and were able to obtain optimal solutions with all IoT devices serviced within the tolerance of the time. The results including some of the decision variables and the optimized objective function are represented in Table 2.

TABLE 2. RESULTS INCLUDING DECISION VARIABLES AND OBJECTIVE FUNCTION

Objective Function Value (Total Traveled Distance)	96,958,431
Total number of Local Servers Deployed	9
Overall Cost	$238,410
Number of Local Server by type (M1, M2, M3)	(8, 0, 1)

VI. SENSITIVITY ANALYSIS

In this part, we varied some of the parameter values in order to show how sensitive the model was to the parameters and extract managerial insights.

Tables 3 represents the sensitivity of objective function values to different satisfaction levels α for probabilistic constraints in the model. It shows that a larger probability of satisfaction leads to more limitation in feasible solution space and can worsen the objective function value (longer total travelled distance) with a budget constraint. It can be concluded that if we consider a deterministic model with 100 percent satisfaction for constraints, the value of the objective function will be worse.

TABLE 3. DIFFERENT SATISFACTION LEVEL (α) FOR PROBABILISTIC CONSTRAINTS

Alpha (α)	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.99**
Objective Function Value (Total Traveled Distance)	96,664,592	96,872,642	96,900,874	96,936,467	96,958,431	97,123,003

Tables 4 represents the sensitivity of objective function values to different values of the maximum tolerable response time T. Results indicate that tightening the tolerable response time would increase the number of local servers, especially the less expensive ones, being deployed. This was largely because a shorter maximum tolerable response time constraint would require servers installed closer IoT devices. This could lead to more servers installed closer to IoT devices in order to meet this constraint. This result could help decision makers to be properly prepared with budgeting and procurement of servers to meet the demand with different response time.

TABLE 4. DIFFERENT MAXIMUM TOLERABLE RESPONSE TIME T ON COST

Maximum Tolerable Response Time (Microseconds)	T=2	T=3	T=4	T=6	T=10
Objective Function Value (Total Traveled Distance)	**98,341,609**	**96,958,431**	**95,726,125**	**93,765,135**	**92,743,243**
Number of Local Server by type (M1, M2, M3)	(12, 1, 0)	(8, 0, 1)	(7, 0, 1)	(3, 0, 1)	(0, 0, 1)

Tables 5 represents the sensitivity of objective function values to different traffic density of a metropolitan area with a fixed maximum tolerable response time T of value 8 and demand. We created 5 levels of density by multiplying the original distances with factors ¼, ½, 1, 2, 4. Results indicate that when the same amount of traffic was generated in a smaller area, there were fewer local servers deployed than the ones for a larger area. This was largely due to the fact that larger area with longer distance between servers and IoT devices would require more servers installed near IoT devices to meet the maximum tolerable response time constraint. This result could help decision makers to be properly prepared with budgeting and procurement of servers to meet the demand based on the size and density of their cities.

TABLE 5. TRAFFIC DENSITY ON COST

Maximum Tolerable Response Time	T=8				
Distance	1/4*D	1/2*D	D	2*D	4*D
Objective Function Values (Total Traveled Distance)	**23,239,739**	**46,407,574**	**92,743,243**	**191,195,451**	**392,063,614**
Number of Local Server by type (M1, M2, M3)	(0, 0, 1)	(0, 0, 1)	(0, 1, 1)	(7, 0, 1)	(16, 0, 0)

VII. CONCLUSIONS

In this paper, we proposed a decentralized server system to properly manage and reduce response time in an intelligent traffic information system. In such a system, numerous IoT devices are installed on the side of the road to interact with vehicles in traffic throughout the entire metropolitan area. Multiple local servers are strategically located in different areas close to the IoT devices. These local servers collect and process data from nearby IoT devices and provide speedy feedback for traffic and vehicle guidance. At the same time, the local servers also serve as intermediaries to communicate with a central server for global traffic controls in the city. We developed a binary nonlinear constrained programming model to minimize overall response time by optimally locating local/intermediary servers and a central server with budget constraint and maximum response time constraint to serve all IoT devices connected on the network. We used MATLAB R2019a MINLP to solve the problem. We also developed multiple sensitivity analysis on factors including stochastic constraint satisfaction, maximum tolerable response time, and traffic density to help with managerial implications of the model and help cities better allocate resources to meet the traffic demand.

IoT and traffic information system as a technology and server location problems are still in their infancy and, thus, a great deal remains for further development and refinement of their modeling and solution methodologies. Our model assumed all clients were covered. There can be several natural extensions to this model. First, we can study a market in which a service provider seeks profit maximization by deciding which client to serve and how much to serve. Second, we can study a competitive market where there could be two service providers competing for clients. In addition, this research utilized generated data. We created a simulated environment to approximate the real-life data. Nevertheless, the study could certainly benefit from using real-life data to validate the model. Empirical research can help establish with what aspects of the traffic controls cities will be most concerned. It can also help with better understanding of the qualities of the available services.

REFERENCES

Allström, J., Barceló, Ekström, J., Grumert, E., Gundlegård, D. and Rydergren, C., Traffic Management for Smart Cities, In Angelakis, V., Tragos, E., Pöhls, H.C., Kapovits, A., Bassi, A. (eds.), Designing, Developing, and Facilitating Smart Cities: Urban Design to IoT

Solutions, Springer International Publishing, 2016, 211-240.

Al-Sakran, H., “Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology”, International Journal of Advanced Computer Science and Applications, 5(2), 2015, 37-43.

An, S., Lee, B. and Shin, D., ‘A Survey of Intelligent Transportation Systems’, *Proceedings of the Third International Conference on Computational Intelligence, Communication Systems and Networks, (CICSYN '11).* IEEE Computer Society, Washington, DC, USA, 2011, 332-337.

Avasalcai, C. and Schahram, D., ‘Latency-aware Decentralized Resource Management for IoT Applications’, *Proceedings of the 8th International Conference on the Internet of Things (IOT '18).* ACM, New York, NY, USA, Article 30, 2018.

Flavio, B. and Milito, R., ‘Fog computing and its role in the internet of things’, *Proceedings of the first edition of the MCC Workshop On Mobile Cloud Computing*, ACM, New York, NY, USA, 2012, 13-16.

Hassanlou, K. “A Multi Period Portfolio Selection Using Chance Constrained Programming”, *Decision Science Letter*, 6, 2016, 221–232.

Holland, J.H., *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.

Johansson, J.M., “On the Impact of Network Latency on Distributed Systems Design”, *Information Technology Management*, (1), 2000, 183-194.

Kevin, A., "That 'Internet of Things' Thing", *RFID Journal*, 22 June 2009, https://www.rfidjournal.com/articles/view?4986 (accessed Nov 1, 2019).

Lu, H., Liu, Q., Tian, D., Li, Y., Kim, H. and Serikawa, S. “The Cognitive Internet of Vehicles for Autonomous Driving,” IEEE Network, 33(3) 2019, 65-73.

Liu, B., *Theory and Practice of Uncertain Programming*, 3rd ed., UTLAB, 2009.

Nemtanu, F.C., Dumitrescu, C., Banu, C.V. and Banu, G.S., ‘Monitoring System with Applications in Road Transport”, *Proceedings of the 20th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Bucharest, 2014, 145-148.

Ramazani, A. and Vahdat-Nejad, H., ‘A New Context-Aware Approach to Traffic Congestion’, *Proceedings of the 4th International Conference on Computer and Knowledge Engineering (ICCKE)* Mashhad, 2014, 504-508.

Raptis, T.P., Passarella A. and Conti, M., ‘Maximizing Industrial IoT Network Lifetime under Latency Constraints Through Edge Data Distribution’, *Proceedings of the IEEE Industrial Cyber-Physical Systems (ICPS),* St. Petersburg, 2018, 708-713.

Rath, M., ‘Smart Traffic Management System for Traffic Control using Automated Mechanical and Electronic Devices’, *IOP Conference Series: Materials Science and Engineering*. 377, 2018

Sherman, F., “Network Latency Milliseconds Per Mile”, https://www.techwalla.com/articles/network-latency-milliseconds-per-mile (accessed Nov 1, 2019).

Shi W. and Dustdar, S., “The Promise of Edge Computing”, *Computer* 49(5), 2016, 78–81.

Sukode, S., Gite, S. and Agrawal, H., “Context Aware Framework in IoT: A survey”, *International Journal of Advanced Trends in Computer Science and Engineering*, 4, 2015, 1-9.

Sun, Y. and Hassanlou, M., “Equity Trading Server Allocation Using Chance

Constrained Programming", *Journal of Supply Chain and Operations Management*, 17(1), 2019, 1-13.