Efficient Urban Transportation(s) with IoT Devices and Robust Workers Allocation

Leonidas Akritidis, Panayiotis Bozanis
Electrical and Computer Engineering Dept., DaSE Lab
University of Thessaly
Volos, Greece
{leoakr, pbozanis}@e-ce.uth.gr

Dimitrios Papadopoulos
Anything Research
Thessaloniki, Greece
dhspapa@anything.gr

***Abstract*—During the past few years, the mobile devices and wireless sensors have played an increasingly important role in the evolution of transportations. These devices are able to collect a wide variety of data about a transporter (such as its location, speed, bearing, fuel consumption, vehicle state, etc) and transmit it via the Internet to special processing nodes. Several methods which exploit this data have been proposed in the current relevant literature. The objective of these works is to improve the efficiency and the economical impacts of the transportation systems. In this paper we present *Anything*, a system designed to facilitate robust and efficient deliveries within the geographical boundaries of a city. We show that by designing a special centralized system which communicates with numerous transporters equipped with online IoT devices, and by employing an effective job assignment algorithm, we are able to achieve fast urban transportations of sensitive packages within a time frame of approximately one hour.**

Keywords—transportations; Internet of Things; job assignment; wireless sensors; mobile devices

I. Introduction

Nowadays, computers are not the only devices which allow connection to the Internet. Mobile phones, tablets, televisions, wireless sensors, vehicles, smart homes and numerous other types of devices allow the connection and data exchange through the Internet. The networks of these devices are now broadly known as the Internet of Things (IoT). According to [1], more than 8.4 billion online capable devices existed in 2017, and more than 30 billion objects will comprise IoT by 2020.

IoT devices are playing an increasingly important role in the improvement of efficiency, accuracy and economic benefit of various systems. Transportations is one of the primary sectors which exploited the concepts of the IoT technology. Wireless sensors, GPS enabled devices, and real-time monitoring systems lead to the decrease of the delivery times, reduced costs, optimal route planning, traffic congestion detection and avoidance, and multiple other factors which collectively resulted in the introduction of the research field of the Intelligent Transportation Systems (ITS).

The significant role of the IoT devices in the modern ITS has been the main research objective for multiple researchers. In [2] the authors examine the effect of these devices in the efficiency of the public transportation systems. Two complete and systematic surveys of the state-of-the-art approaches, implementations, solutions, and technologies across a broad range of projects for vehicular communication systems are [3] and [4]. Another survey examines similar issues and applications regarding the Parallel transportation Management Systems [5]. Moreover, the management of incidents in intelligent transportation systems is the objective of [9]. In this work, several models and systems which exploit real-time traffic data are reviewed.

The management of the data generated by the IoT devices and the wireless vehicular sensors has led to the Data Driven Intelligent Transportation Systems (D^2ITS). In [7] the authors discuss the functionality of the key components of these systems. An interesting problem regarding the data generated by multiple different sources, is their combination with the aim of obtaining better inference. Data Fusion is a collection of techniques which achieve this combination of information from diverse data sources [6,8].

In this paper we describe ***Anything***, an urban transportation system which is based on such IoT devices. In contrast to other existing approaches, *Anything* is specially designed to facilitate urgent transportations within the geographical boundaries of a city. We call a transportation urgent, if the requirement of the sender is to have his/her package/s delivered to the recipient/s, within a very short period of time (e.g. one, or even half an hour). Examples of such transportations include deliveries of food products, important documents, presents (e.g., flowers) and other types of sensitive packages.

The system we present here accepts transportation requests (called *jobs*) by either individuals or companies (called *employers*) through a simple Web or Android interface. In the sequel, it searches a pool of available co-operating transporters (called *workers*) and identifies the most suitable one to execute the requested job. The platform communicates with the available workers via standard mobile devices (phones, tablets, etc) which are equipped with a standard GPS receiver and have an active Internet connection. The identification of the most suitable worker is performed by taking into consideration various parameters, including the distance, the priority of the employer, the type of the package to be delivered, the current delay in the delivery, and numerous others. All these parameters are embodied within a cost function which is computed for each available worker before a job is assigned. After a job has been assigned, the worker departs and moves towards the location of the employer where s/he will receive and load the packages to be delivered. In the sequel, these packages are sequentially delivered to their final recipients.

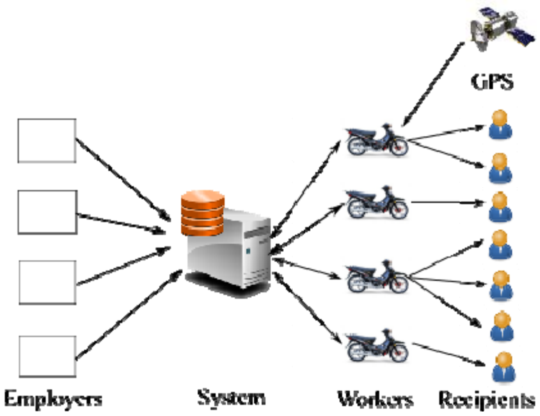

Fig. 1. High level architecture of the *Anything* transportation system

The system design includes advanced features regarding fault-tolerance. Throughout the entire lifecycle of a transportation job, the worker may encounter non-recoverable incidents such as mechanical failures or accidents. For this reason s/he is always given the choice to abandon or cancel a job. In such cases, the platform will attempt to re-assign the job to another worker. In comparison to other transportation systems, the issue of fault-tolerance is generally overlooked and more specifically, the problem of automated recovery from such situations has not been studied extensively.

The rest of the paper is organized as follows: In Section III we provide a detailed description of how a transportation is executed with the *Anything* system, whereas in Section IV we discuss the communications of the system with the employers and the workers. In addition, the method which assigns jobs to the workers is described in Section V. Finally, in Section VI we present an evaluation of the system whereas in Section VII we state our conclusions.

II. Description of the Transportations Workflow

In this Section we describe the main components of the *Anything* transportation system, and the procedure of a job execution.

Figure 1 illustrates a high level diagram of the involved communications. The system accepts messages from a set of W workers, that is, transporters who are capable of executing transportation jobs. In the sequel, it processes these messages and produces the appropriate responses. Notice that each worker can only execute one job at a time and cannot perform multiple actions simultaneously. In addition, the system offers a simply designed interface to the E employers to allow them submit their transportation jobs.

At any moment, the system has a complete and accurate picture of all of its co-operating workers. The three primary phases of a transportation execution are graphically illustrated in Figure 2. When an employer e submits a new transportation job, an assignment module within the platform first searches for existing available workers. In case it cannot find one (e.g., all workers are executing other jobs at that time), it enlists the job in a priority queue with the aim of processing it when a worker becomes available again. In the opposite case, it applies the job assignment algorithm (described in Section IV) to select the most appropriate worker for the job execution.

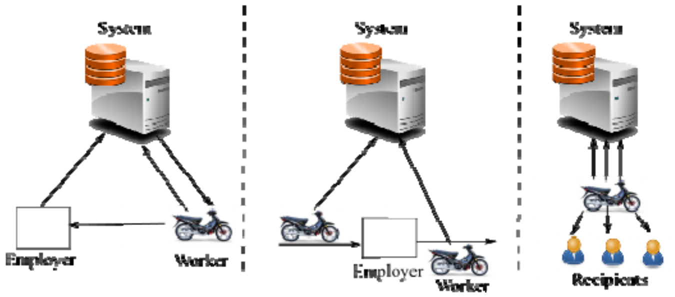

Fig. 2. The three primary phases of the execution of a transportation: During the first phase (left), an employer requests for a worker. The system searches the pool of the available workers and assigns the job to the most suitable one. In case the worker accepts the job, s/he transmits an acceptance message and departs for the location of the employer. During the second phase (middle), the worker arrives to the employer, receives the packages and departs to deliver them to their recipients. The system is notified for both the arrival and the departure from the employer. Finally, in the third phase (right) the worker delivers the packages to their respecitve recipients and notifies the system for each successful delivery. The job is completed when the last package is finally delivered.

A worker w can be in one of the following three possible states:

- *Active and available*: A new job can be assigned to w.
- *Active and unavailable*: The worker w is currently executing another job, consequently, s/he cannot accept a new one.
- *Inactive*: The worker w is not executing another job, however, s/he is not in condition to accept a new one. This category includes the workers who have a temporary or permanent problem (such as mechanical failure, malfunction, accident, etc.), or are simply out of duty.

After the assignment of a job j, the selected worker w has the option to either accept it, or reject it. The rejection of an assignment means that w has a experienced an important incident and cannot execute any jobs. Therefore, the system automatically deactivates w and attempts to assign the job to another worker w'. The worker who rejects a job shall remain inactive, until the problem s/he encountered is addressed. In such occasions, the reactivation of a worker can only be performed manually by specialized personnel who handles the platform at that time.

In case w accepts the assigned job j, s/he departs from his/her present location and moves towards the employer e to receive and load the package/s. The system removes w from the pool of the available workers, and updates his/her status to

"*Active and unavailable*". From this point on, and as long as s/he executes *j*, *w* will not be able to accept any new jobs. If during the transition to *e* an important incident occurs, *w* is given the opportunity to abandon the job. As a result, the system will deactivate *w* and assign *j* to another available and active worker *w′*. In case no problem occurs, *w* will eventually arrive to *e* to receive and load the packages which will be transported to their final destinations.

The employer provides the worker with all the necessary details of the job (i.e. recipients' addresses, special demands, price, etc). Notice that a single transportation job may involve deliveries to multiple recipients. In this case, the job is split to a number of tasks, which is equal to the number of the recipients. To maximize the efficiency of the platform, it is required that the time that *w* stays stationary in the location of *e* is minimal. For this reason, the employers are instructed to have their packages and the respective details ready by the time *w* arrives at their location. The platform calculates and stores a number of useful statistics including the duration that each worker stays at each employer. These statistics are later analyzed and such bottlenecks in a transportation are detected and eliminated.

The worker uses the mobile application to notify the system for the arrival to and departure from *e*. In addition, *w* transmits the locations of the recipients and the number of tasks to be fulfilled. In the sequel, s/he successively executes each task by completing the required deliveries. The last delivery signifies the successful completion of the entire transportation job. When that signal is received, the system updates the status of the worker to "*Active and available*" and instructs him/her to move to the predefined starting point. From this moment the worker becomes available to accept new jobs again.

Finally, notice that an unrecoverable incident may occur during the transportation of the packages to their destinations. In such occasions, the worker *w* notifies the system which in turn deactivates *w* and assigns the job to another available and active worker *w′*. Nevertheless, since the packages are not in the location of the employer any more, *w′* is instructed to go to the location of *w* and receive the remaining packages to be delivered.

III. COMMUNICATIONS

In this Section we describe how the system communicates with the employers and the workers.

A. Employers-System Communication

The communication between the employers and the system is unidirectional and very simple (see Figure 1). A small Web or mobile application is installed in a computer or mobile device in the location of each employer. When there is a transportation job to be executed, the employer requests for a worker simply by pressing a button in the installed application.

This action automatically creates a new transportation job in the system and the procedure which we discussed earlier in Section II starts.

TABLE I. MESSAGES GENERATED BY THE WORKERS

Type	Description
1	Request for a new job
2	Job accepted
3	Job rejected
4	Job completed
5	Job cancelled before the receipt of the package/s from the employer (Job cancellation)
6	Job cancelled after the receipt of the package/s from the employer (Job abandonment)
7	Arrival at the employer
8	Departure from the employer
9	Task completed
99	Transmit DeviceID and GPS data

B. Workers-System Communcation

In contrast to the previous case, the communication of the workers with the system is bidirectional and considerably more complex. Each worker is supplied with a standard mobile device (such as a phone or a tablet) which is equipped with a standard GPS receiver and an active connection to the Internet. An application which runs on the device periodically transmits to the system a variety of geo-spatial data including the location of the worker, its speed, bearing and altitude. It also transmits additional data regarding the worker's availability and status. The data is transmitted via the Internet connection of the mobile device, and it is submitted to special Web location maintained by the platform.

This application allows the worker to contact the system by pressing buttons (which reflect an action), or making selections from lists (e.g., list of areas in the city, or lists of tasks). Furthermore, it implements advanced fault-tolerant characteristics. Fault-tolerance is crucial, since a variety of unexpected conditions may occur during a transportation. For instance, there are cases where the Internet connection and/or the communication with the GPS satellites are not always available (e.g., when the worker enters an elevator or a basement). In such cases, the application caches both the GPS data and the actions of the worker and transmits them to the system when the Internet connection becomes available again. Another type of unexpected condition is the occurrence of a sudden incident with the worker (for example, mechanical failure, accident, etc.) which rendered him/her incapable of completing a job. For this reason, the application always allows the worker to reject, abandon or cancel a job.

The system maintains a typical Web server which accepts requests from the workers in the form of a typical Web request of GET type. These requests can either be explicitly submitted by a worker during an action, or periodically generated by the application which runs on the mobile devices of the workers. As we will see shortly, the requests can be of several types. The server forwards each request to the internal modules of the platform which process it and generate an XML-formatted response. This response is then received by the aforementioned application; the XML content is parsed by a built-in response analyzer and user-friendly messages appear on the screen of the device, or pre-recorded sounds are played, or a location on a map is drawn.

Here we note that each mobile device of each worker possesses a unique internal device identifier (DeviceID). This DeviceID accompanies every communication of the worker with the platform and it is used for identification purposes. More specifically, there is a special database table which stores the DeviceIDs of all workers. When a worker attempts to contact the system, a validation process checks the validity of the received DeviceID by querying this table. In case of successful validation, the request is processed; otherwise, the request is rejected and a relevant message appears on the screen of the device.

In Table I we show the ten different types of Worker-System communications. Type 99 refers to the case where a worker simply transmits GPS-related data (i.e., location, speed, bearing and altitude). In fact, this kind of data is also sent in all the other communication types. In other words, all transmissions to the platform include a type 99 message.

Requests of type 1 are submitted only by workers who are both active and available. In case such a request is submitted to the platform, the assignment module searches the queue of pending jobs and applies the assignment algorithm of Section IV. In the sequel, in case it decides that a job should be assigned to the worker who makes the request, it transmits the respective details (location of the employer, etc.). Otherwise, it simply responds by sending a "No pending job" message.

The request types 3, 5, and 6 notify the system that a worker has a serious problem. These three communication types lead to the deactivation of the worker and the assignment of a job to another worker. Notice that a worker can only be re-activated manually, by the personnel who monitors the system, after the problem is solved.

Finally, the request types 2, 7, 8, and 9 are sent to notify the system about the states of a worker (location, status, etc.) and the assigned job (task or job completion). Notice that a request of type 2 changes the status of a worker from available to unavailable, whereas the opposite happens for the type 9 requests.

IV. JOB ASSIGNMENT ALGORITHM

Now let us describe the algorithm which determines the assignment of a job to one of the available workers.

A. Cost Matrix

Let e be an arbitrary employer who submits a job $j \in J$ where J is a set which contains all the current unassigned jobs. Each employer e is assigned a priority value p_e which determines how important the jobs submitted by e are. For instance, the employers who request transportation of fresh food are assigned a high priority value, whereas the employers who request document deliveries have a lower priority. Moreover, we introduce the set W which contains all the active and available workers who are currently in the condition to accept and execute a new job.

Now for a job j and an available worker w we introduce the cost function $c(j,w)$ which represents the cost of execution of j by w. In the next subsection we shall provide the details about the parameters of this cost function and how it is implemented in the *Anything* transportation system.

In the general case where $|W|$ is the number of the available workers and $|J|$ is the number of the pending jobs, we define the following general $|J| \times |W|$ cost matrix:

$$C = \begin{bmatrix} c(j_1, w_1) & c(j_1, w_2) & \cdots \\ c(j_2, w_1) & c(j_2, w_2) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{1}$$

The construction of this matrix implies that for each pending job, we must compute one assignment cost per worker. Our goal is to identify the worker who is the most suitable to execute each pending job. Therefore, for each row of this matrix we compute all the costs in each column by employing the aforementioned score function.

In the sequel, the jobs are assigned in increasing cost order. More specifically, the minimum element in this matrix $\min\{C\}$ indicates a (j_x,w_y) pair. The algorithm assigns j_x to w_y and removes the x^{th} row and the y^{th} column. This procedure is repeated until all jobs are assigned to a worker, that is, the matrix has no more rows left, or when the available workers are exhausted and, thus, no more columns are left.

Now we distinguish two cases:

- $|W| \geq |J|$: In this case the number of the available workers are greater than or equal to the number of the pending jobs. Therefore, we are able to assign all jobs and empty the respective queue. Of course, a number of workers will not be assigned a job.
- $|W| < |J|$: This is the opposite case, where there are not adequate workers to execute all pending jobs. All workers will be assigned a job, but unavoidably, some jobs will remain in the queue waiting for new workers to be released. As we describe in the next Subsection, the jobs which remain for a long time in the queue obtain a higher priority. Therefore, the probability that they will be assigned sooner (in comparison to jobs which have recently entered the queue) increases.

B. Cost Function

The $c(j,w)$ elements of the cost matrix of (1) are essentially cost functions which quantify the cost of assigning an arbitrary job j to a worker w. This function must implement the goal of identifying the worker w who is the most suitable to execute a job for an employer e. In the *Anything* transportation system, the employed cost function has the following form:

$$c(j, w_i) = k_1 d(e, w_i) + k_2 p_e (T - (t - t_j)) \tag{2}$$

where k_1 and k_2 are two predefined constant quantities which satisfy the following limitation:

TABLE II. STATISTICS REGARDING THE OPERATION OF THE SYSTEM

Statistic	**Value**
Number of jobs	907
Number of tasks	1358
Average tasks / job	1.497
Average covered distance / worker	3487 km
Average time to accept / reject a job	00:00:14
Average time to travel to the employer	00:15:36
Average time in the location of the worker	00:04:13
Average delivery time (per job)	00:36:50

$$k_1 + k_2 = 1 \qquad (3)$$

The term $d(e,w_i)$ in (2) represents the distance between e and w_i. Given that φ_e and λ_e are the latitude and the longitude values of the employer (which are stable), and φ_w and λ_w are the respective latitude and longitude values of a worker (which may change), the distance between them is approximated by the well-known haversine formula:

$$d(e,w) = 2R\arctan 2(\sqrt{a}, \sqrt{1-a}) \qquad (4)$$

where R is the radius of a spherical Earth (approximately 6400 km), and:

$$a = \sin^2 \frac{\phi_e - \varphi_w}{2} + \cos\phi_e \cos\phi_w \sin^2 \frac{\lambda_e - \lambda_w}{2} \qquad (5)$$

Moreover, in the right part of (2) we encounter t, which is the current timestamp, t_j which represents the time of submission of j to the system, and T, which is a constant quantity that symbolizes the maximum acceptable delay for any job.

Notice that (4) does not take into consideration the street design of the city, nor the route that the worker needs to follow to reach the employer. Furthermore, it is insensible to the traffic congestion. These are issues which we plan to deal with in the near future.

To conclude, the notions which are implemented by the cost function of (2) are:

- To assign a job to the worker which is closest to the employer. Therefore, the higher the distance, the higher the assignment cost is. In other words, we attempt to minimize the cost of the transition of a worker to the employer.
- To prioritize the jobs according to the delay $t-t_j$. As the time passes, the term $t-t_j$ increases; therefore, $T-(t-t_j)$ decreases. Equivalently, as the time passes, the assignment cost of a job is reduced.
- To prioritize the jobs according to the employer. This is achieved by the factor p_e which characterizes each employer.

TABLE III. JOB DISTRIBUTION PER EMPLOYER TYPE

Employer Type	**Number of Jobs**
Hot food (B2C)	261
Beverages	203
Cold food (B2C)	187
Generic food products (B2B)	154
Flowers	22
Presents	50
Documents	26
Others	4

V. SYSTEM PERFORMANCE

The *Anything* transportation system operates in Thessaloniki, the second greatest city in Greece. Currently, it co-operates with seven workers who move across the city by mainly using motorcycles. Moreover, *Anything* serves systematically more than 70 enterprises-employers, as well as numerous individuals. The enterprises are of various categories: food suppliers, restaurants, flower and present shops, candy stores, liquor stores, super markets, and numerous others.

As discussed earlier, an employer who desires to send one more packages to a number of recipients can either use an Android app, or a simple Web interface. Both applications implement a simple mechanism for calling a worker. On the other hand, the workers possess a mobile device which runs a more sophisticated application whose operation has been described earlier in Section III. A special information system stands between the employers and the workers and it is primarily responsible for accepting the requests of the employers, the assignment of jobs to the workers, the real-time monitoring of the transportations, and the recording of various statistics, such as delivery times, covered distances, location history, etc. This system also keeps track of the charges and other financial information, which, however, is out of the scope of this paper.

The application of the workers periodically contacts the aforementioned information system. The length of that period has been set to 10 seconds. The reasons for this selecting contact periods of 10 seconds is that i) shorter periods have a negative impact on the battery of the mobile devices of the workers (a powerbank is required to keep alive the device for more than 8 hours); and ii) longer periods result in an inaccurate picture of the current location of a worker.

In Table II we provide several interesting statistics regarding the system performance. The values we present here concern the first month of 2018. In that period, there were in total 907 transportation requests submitted to the system. The total number of the assigned tasks (i.e. deliveries to different recipients) was 1358, which leads to an average of 1.5 task per job. The workers covered, on average, 3487 kilometers during this period. This is translated to an average daily covered distance of 112.5 km.

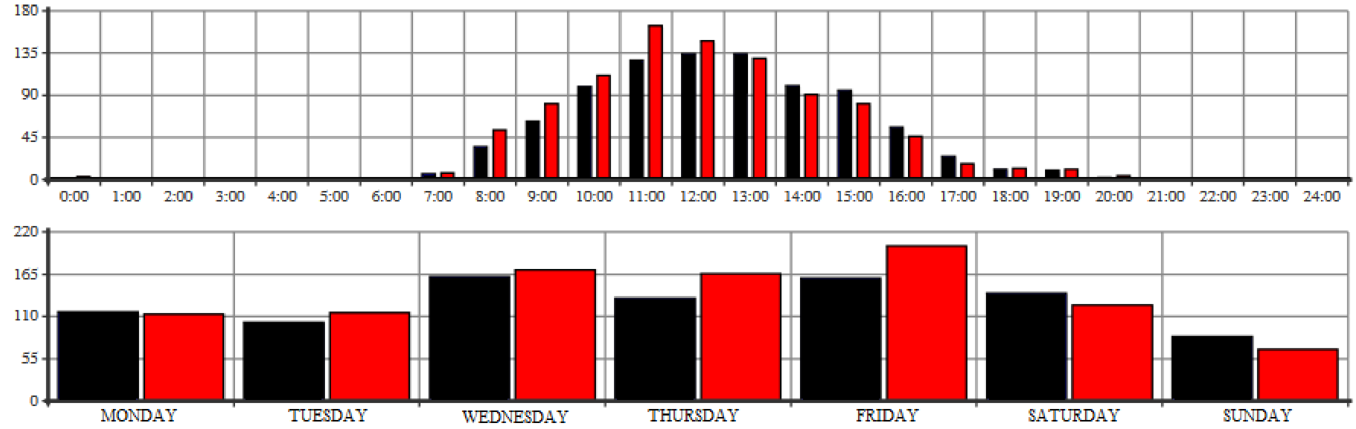

Fig. 3. Distribution of jobs submitted to *Anything* between 01/01/2018 and 31/01/2018 by i) time (upper diagram); and ii) by day of week (lower diagram)

The average job completion time was about 55 minutes. This duration is analyzed to the time the workers consumed to accept or reject a delivery (11 seconds on average), the average duration of the transition to the location of the employers (15 minutes and 36 seconds), the time that the workers stayed stationary at the employers to load the packages (4 minutes and 13 seconds), and, lastly, the duration of the delivery of the packages to their recipients (on average 36 minutes and 50 seconds).

Moreover, in Table III we present the distribution of the 907 jobs per employer type. The majority of the transportations included deliveries of food products. More specifically, the workers of Anything completed 261 and 203 deliveries of hot and cold food products respectively. The recipients of these products were all individual customers (B2C). The workers also performed 154 deliveries of food products from the employers to other enterprises (B2B). Other types of frequent transportations included beverages (such as hot and cold coffees, tea, drinks, etc. - 203 jobs), presents (toys, jewelry, clothing and shoes - 50 jobs), and flowers (22 jobs).

Finally, in Figure 3 we illustrate the distribution of the jobs submitted to *Anything* between 01/01/2018 and 31/01/2018 by time (upper diagram), and by day of week (lower diagram). The red bars represent the number of jobs, whereas the black bars represent the number of tasks. We observe that the peak of the system occurs on average around 11 o'clock ± 2 hours. Moreover, the busiest day on average is Friday, followed by Wednesday.

VI. Conclusions

In this paper we presented *Anything*, an urban transportation system which is designed to allow package deliveries with three special requirements: i) the sender and the recipients are within the geographical boundaries of the same city; ii) the deliveries must be completed in a very short time window of about 1 hour; and iii) fault-tolerance, i.e., to effectively overcome unpredictable incidents, such as mechanical failures and accidents, in an automated manner. *Anything* achieves all three requirements by employing mobile devices with standard GPS receivers and an active connection to the Internet. With careful job assignment and robust workers allocation, the presented system allows fast and cost-effective transportations of sensitive packages, like food products, important documents, presents, etc.

It is within our intentions to further develop the Anything transportation model by performing research about issues such as the detection and avoidance of traffic congestion, optimal route planning, dynamic rerouting, and comparison of the theoretical background with other existing approaches. We also intend to introduce additional IoT devices and sensors with the aim of monitoring crucial parameters and statistics, including the fuel consumption, the generic condition of the vehicles, and the fluctuation of the packages temperature during transportation.

References

[1] https://en.wikipedia.org/wiki/Internet_of_things

[2] M. Al-Rousan, AR Al-Ali, K. Darwish. GSM-based Mobile Tele-monitoring and Management System for Inter-cities Public Transportations. Proceedings of the 2004 IEEE International Conference on Industrial Technology, pp.859-862, vol. 2, 2004.

[3] L. Figueiredo I. Jesus, J.A.T. Machado, J.R. Ferreira, J.L. Martins De Carvalho. Towards the Development of Intellignet Transportation Systems. Proceedings of the 2001 IEEE Conference on Intellignet Transportation Systems. pp 1206-1211, 2001.

[4] P. Papadimitratos, A. De La Fortelle, K. Evenssen, R. Brignolo, S. Cosenza. Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation. IEEE Communications Magazine, 47(11), 2009.

[5] F.Y. Wang. Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications. IEEE Transactions on Intelligent Transportation Systems, pp. 630-638, vol. 11, no. 3, 2010.

[6] Y. Zhao. Mobile Phone Location Determination and its Impact on Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, pp. 55-64, vol. 1, no. 1, 2000.

[7] J. Zhang, F.Y. Wang, K. Wang, W.H. Lin, X. Xu, C. Chen. Data Driven Intellignet Transportation Systems: a Survey. IEEE Transactions on Intelligent Transportation Systems, pp. 1624-1639, vol. 12, no. 4, 2011.

[8] N.E. El Faouzi, H. Leung, A. Kurian. Data fusion in intelligent transportation systems: Progress and challenges--A survey. Information Fusion, pp. 4-10. vol. 12, no. 1, 2011.

[9] Ozbay, Kaan and Kachroo, Pushkin. Incident management in Intellignet Transportation Systems, Artec House Publishers, 1999.