Distributed Backpressure Routing and Byzantine Generals Fault Detection for Electric Vehicles

Evangelos D. Spyrou*, Dimitrios K. Mitrakos

Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Egnatia Odos, Panepistimioupoli, Thessaloniki, Greece

Copyright ©2017 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract Electric vehicles (EVs) emerged in the transport domain, due to their energy efficiency and clean energy that they utilise. The electric vehicle routing problem is essentially a problem of selecting a set of minimum cost routes, while the demand of the customers is achieved. In this work, we model the electric vehicle routing problem using a wireless network methodology, namely the backpressure framework. Every route is imposed with a penalty, which includes the driving time of each road. We derive a weight as a function of the road queue backpressure and the driving time of a car. The next route for our EV is the one that has the highest weight. We show that this methodology leads to faster routes in that there are often roads with accidents or traffic jams. Also, we propose a fault detection mechanism that will ensure that there will be not deficiency in the electric vehicle routing process, in terms of wireless communication. We employ the Byzantine Generals algorithm to detect possible faulty wireless mediums in cars or traffic lights. We show how our approach is capable to detect faulty wireless mediums and we provide an alternative on if the consensus cannot be satisfied.

Keywords Electric Vehicles, Routing, Backpressure Weight, Lyapunov, Fault Detection, Byzantine Generals

1 Introduction

The use of electric vehicles (EVs) into the transport sector has been introduced [30] for two reasons. Firstly, there is the need for energy efficiency in transport, since the charging of the EV is not expensive as opposed to the internal combustion engine vehicles. Secondly, there is the reduction of the CO_2 emissions due to the clean sources (battery) employed for the production of electricity. On the other hand, traditional vehicles are associated to the combustion of fossil fuels. Thus, EVs provide a great alternative as a next generation of the transport means in a city.

The main characteristics of EVs play a key role on the design of algorithms utilized by route planners. EV route

planning exhibits certain differences from conventional route planning. Initially, the limited capacity of the EV's battery introduces the constraints of driving ranges of approximately 100 km on average. Moreover, charging of EVs take place in stations that do not exist in as many places as gas stations. EV charging is a process that may take hours to complete. Thus, routes may be determined in an economic manner rather than just the fastest or shortest. On the other hand, there are cases where the state of the battery does not necessarily decrease when driving. All of the above sum up to the fact that companies with EV fleets, require their goods to be delivered on time in often jammed road networks with potential uncertainties, such as road accidents or road works.

The Vehicle Routing Problem (VRP) was proposed in [27]. Thereafter, a plethora of extensions of the problem have been suggested, which included real world constraints. Two works that constitute the most widely investigated extensions are the Capacitated VRP (CVRP), where vehicles have a limited freight capacity and the VRP with Time Windows (VRPTW), where customers have to be reached within a specified time interval [33] [35]. The Electric Vehicle Routing Problem (EVRP) [41, 21] has been formulated as a problem of locating a set of minimum cost routes, in order for the demand of the costumer to be accomplished. Often, constraints related to the capacity of the battery of the EV have been investigated.

Wireless communication, such as 3G or Wi-Fi, emerged in vehicle-to-vehicle communication. In particular, EV actions such as battery charging information [4, 5]. perform car routing depending on intelligent charging strategies; thus creating Vehicular Ad-hoc Networks (VANET)s. These networks make the decisions themselves in the case of autonomous cars [6] or they assist the drivers in a plethora of functions.

We address the EVRP, where the cars communicate in a wireless manner with other cars and traffic lights to decide on the next routing option that is decided in a distributed manner, as it is described in [8]. However, there might be faults in the car-to-car communication or in the car-to-traffic light communication. Hence, a mechanism is essential that will be able to identify possible faulty wireless modules that are responsible for the successful communication and optimal routing.

Other works exist that aim at the fault detection of various systems in electric vehicles [9, 10, 10, 11, 12, 14, 15].

In this paper we deal with dynamic traffic routing for electric vehicles. For our approach, we employ a methodology utilized in wireless networks, called the Backpressure routing, which essentially builds the routing without routes [34]. The aim is to employ infrastructure, such as traffic cameras to estimate the queues created at each traffic light road. This will provide the driving time of each street and we provide a penalty minimization used in a weight to be calculated for each road. We provide results based on simulations to show the efficiency of our approach, which results in the successful delivery of companies' goods on a predefined schedule in a single depot.

We attempt to complete the Backpressure framework proposed in [8], in order to ensure a robust operating of the routing protocol. We employ a very well-known algorithm, the Byzantine Generals algorithm, to provide the system with a distributed way to detect faults in communications.

We show the following:

- EV routing may be modeled with the backpressure framework, which exhibits stable queues and they are used to construct a weight to select the next best route
- Road driving time is a metric that may act as a penalty that is minimized.
- Backpressure application results in fastest routes, avoiding any sudden road condition changes in a real-time fashion.
- There are cases that backpressure outperforms Dijkstra's shortest path algorithm in reaching a destination via EV routing.
- We employ the Byzantine Generals algorithm, to show the fault detection of the wireless infrastructure
- We show the efficiency of our approach with an example scenario.

The paper is structured as follows: Section 2 presents the related work in electric vehicle routing, section 3 provides a description of the queuing networks and the backpressure framework utilized, section 4 explains the penalty to be minimized, section 5 gives a background on the Byzantine Generals algorithm, section 6 gives the simulations of our proposed scheme and section 7 gives the conclusions of our approach.

2 Related Work

In [20], the authors investigate the one-to-many vehicle routing and scheduling problem in EVs. More specifically, problem formulation and constraints in practical scenarios are examined. They highlight that the EVRP is an NP-hard problem and requires significant computational power for the location of near optimal solutions in medium to large

scale scenarios. The authors provide a mathematical formulation to model the EVRP due to its capacity, time window and predefined charging level constraints. Furthermore, EVRP trends are examined providing significant information regarding future research on real world scenarios and approximation algorithms.

In [25], the authors aim to find the optimal route for EVs in a multi-customer scenario considering recharging requirements during the routes. They formulate routing as a Mixed Integer Linear Programming problem. The battery recharging at every station is a variable, since flexible routes are to be guaranteed. The proposed scheme optimizes total travel, waiting and recharging time as well as the number of the EVs utilised. The problem is solved using Variable Neighborhood Search Branching (VNSB) in reasonable computational times.

In [40], the authors introduce the electric vehicle-routing problem with time windows and recharging stations (EVRPTW). The model employs recharging at any of the predetermined stations using a recharging scheme. Furthermore, they consider limited vehicle freight capacities in conjunction with customer time windows, which constitute constraints in real-world transport applications. To solve the aforementioned problem, a hybrid heuristic is presented, which combines a variable neighborhood search algorithm with a tabu search heuristic.

In [28] the authors propose an intention-aware routing system (IARS) for electric vehicles. The system provides the ability to EVs to estimate a routing policy, which minimizes journey time, while keeping track of other vehicles intentions. Considering other vehicles' intentions is significant since the driver may have to charge the vehicle in the journey and queuing time may be large, in case other vehicles select the same stations. Thus, queuing times are predicted based on the intentions of the other EVs.

In [19], the authors attempt to tackle the problem of energy efficient routing for EVs using Particle Swarm Optimization. They also show that EVs route optimization techniques, such as negative edge costs, battery power and capacity limits, as well as vehicle parameters that are only available at query time, make the task of electric vehicle routing a challenging problem.

In [23] the authors investigate route planning applications for electric vehicles. They show that such problems have to consider constraints such as energy consumption. They indicate that recent approaches for EV routing focus on optimizing energy consumption as a single variable. They provide preliminary work towards a holistic framework for computing shortest paths for electric vehicles with limited range. Their scheme comprises driving energy-efficient speed adjustments, realistic modeling of battery charging and the integration of turn costs.

In [43], the authors describe a number of heuristics targeting the vehicle routing problem in which the fleet comprises electric vehicles. These vehicles possess limited autonomic functionality and they are required to visit the charging stations for recharging during their missions. Thus, the routing schedule needs to be determined in conjunction with the

recharged energy and the technology that is used. To this end, constructive and local search heuristics are proposed using a non deterministic Simulated Annealing framework. The results of this paper shows varying instances and performance issues by also analyzing specific parts of the problem.

In [44], the authors describe a generic EVRP where the optimal routing strategy is identified. Routing takes into account the minimal traveling time cost as well as the energy dissipated and the number of electric vehicles used for a task. The novelty of this work resides in the consideration of the load of the vehicle on the energy consumption. A case study in Austin, Texas is described and the primary result is that the vehicle load is not negligible when it comes to the identification of the routing strategy. Furthermore, the authors compare their approach to a conventional diesel truck routing problem. They identified that EVRP has comparable travel time and distance but long en-route re-charging time, which translates into a considerable amount of additional labor cost. Finally they find that the network topology strongly affects the routing strategy.

In [45], identified that the problem with electric vehicles becomes evident in long journeys, or inter-city journeys. The fact that a driver is under anxiety regarding whether she will reach her destination before the battery runs out, is a preventing reason for the disruption of the specific market. To this end, the authors describe a shortest-walk problem to determine the route from a start to a destination. Note that a route might include cycles for battery recharging. IN this problem two scenarios are investigates, namely the travel distance minimization when a stops for recharging is necessary and when a number of stops is specified. Both problems are polynomially solvable and algorithms are provided for solving them. Finally, a route strategy problem that minimizes driver anxiety is presented.

In [46], the authors present an EVRP with Time Windows and Battery Swapping Stations. A fleet of battery powered electric vehicles accomplish their customer duties within the serving region with a fixed battery swapping time. To this end, time windows of the customer needs are considered. The Gurobi optimizer is utilized to extend and verify small instances of the problem. Two models of swapping and recharging are compared using equivalent parameters. Results show that the number of vehicles can be reduced using this approach.

In [47], the authors suggest a queuing framework for the EVRP. They develop routing strategies, which optimize a metric concerning the vehicles' sojourn system time. They incorporate the "Join-the-Shortest-Queue" approach and they encapsulate the necessary weights to maximize the system's throughput. Also, they mitigate a vehicle's sojourn time. Since the strategies adapt to the speed of charging at the stations, this has an effect on the distribution of the vehicles point of origin when requesting service, the traffic congestion level and the vehicle speed.

3 Queuing Networks and Backpressure Framework

Initially we begin this section by giving a background of the M/M/1 queue and we will see how it relates to the backpressure framework.

3.1 M/M/1 Queue

For an M/M/1 queue, we have one junction with an exponential service rate μ . The arrival rate to the junction is $\lambda < \mu$. Moreover, the waiting area can be infinite. We can derive that the average number of cars in the junction is

$$L_s = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho} \quad (1)$$

where $\rho = \lambda/\mu$ is the utilization of the junction. Notably, L_s is denoted as the average number of cars in the entire systems and not the queue. If we use Little's law $L = \lambda W$, we know that the average time a car spends in the junction is

$$W_s = \frac{1}{\mu - \lambda} \quad (2)$$

Thus, W_s includes the time a car spends in the queue as well as the junction. Thereafter, we let L_d and W_d be the average number of customers and the average waiting time in the queue, respectively. We know that the average waiting time in the junction W_s comprises the average time in the queue and the average service time that we denote as W_t as

$$W_s = W_q + W_t \quad (3)$$

where $W_t = 1/\mu$. Hence we have

$$W_q = W_s - W_t = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)} \quad (4)$$

Then by Little's law to the queue we have

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{\rho^2}{1 - \rho} \quad (5)$$

3.2 Routing for Wireless Networks

The inspiration from our work comes from the field of wireless networks, routing in particular. We consider a network of roads where routing decisions are undertaken in a distributed fashion at every traffic light change. Hence, we consider a queue in every road where traffic is bidirectional. Figure 1 shows a basic road networks. In this scenario there are four junctions with a queue of a number of cars at the traffic lights.

We denote the cars that are to forwarded to road c as d . Let $Q_n^{(d)}(t)$ be the current amount of data in node n , which is called the queue backlog. As we see in [16], the road queue backlog from one traffic light change slot to the next satisfy

$$Q_i^{(d)}(t+1) \leq \max[Q_i^{(d)}(t) - \sum_{b=1}^N \mu_{ib}^{(d)}(t), 0] + \sum_{a=1}^N \mu_{ai}^{(d)}(t) + A_i^{(d)}(t) \quad (6)$$

Figure 1. Basic Queuing Network

for all $i \in 1, \dots, N$ such as $i \neq c$. In this equation $A_i^{(d)}(t)$ corresponds to the cars that arrive to road n in the traffic light change slot t . Furthermore, $\mu_{ib}^{(d)}(t)$ is the car forwarding rate on car linking junctions (a, b) on time t . Note that $\mu_{ib}^{(d)}(t)$ can be greater than the cars d that are forwarded between junctions (a, b) on slot t . Moreover, $\sum_{a=1}^N \mu_{ai}^{(d)}(t)$ may be greater than the arrivals of cars to the junction i on time traffic light change slot t . Hence, we have an inequality in 6, since we may have limited backlog in junction a .

The dynamic nature of the road network combined with the unwanted but sometimes occurring bottlenecks, provide the necessary means for the emergence of a dynamic routing algorithm, where routing decisions will be made dynamically each time a car enters a road leading to a junction. Hence, it is quite useful to employ a wireless network routing algorithm that performs this task, with respect to the road network conditions. Modern road networks utilize traffic cameras to monitor traffic and to prevent speeding above the limit. Hence, it is quite reasonable to assume that they may be used to calculate driving time and number of cars within a road. Furthermore, with the emergence of the Internet of Things [32] and Cyber-Physical Systems [22], it is clear that infotainment systems in cars maybe connected to devices on junction traffic lights that may exchange traffic information, resulting thus, to a wireless road network. Hence, it may be promising to employ a dynamic routing algorithm in a road network, if we consider it as a graph-theoretic entity [24] where vertices are considered as junctions and edges as roads.

3.3 Backpressure Routing

Backpressure routing does not operate like traditional routing mechanisms, meaning that it does not locate an explicit path estimation from any source to a destination. It performs routing decisions for each car by calculating for each outgoing road a backpressure weight. This weight is a function of localized queue and link state information. In figure 2 we observe the backpressure functionality in a simple road network. Note that Node B represents an intermediate routing junction.

Routes in backpressure are learned dynamically and maybe different for different cars. The delay incurred maybe

large, especially when the junction is not heavily loaded. Thus, there is not enough pressure to push the car towards the destination. The network of junctions is trivially stable; however, the pressure gradients do not build up. Furthermore, we have to note that there is the possibility of a backpressure implementation that is subjected to some pre-specified routes. This may have a restriction impact on the capacity region; however, it might improve delivery and delay. Another manner that improves delay is the use of backpressure version implemented in [42], which adds a "differential progress-to-destination" term to the backpressure index.

We will next provide a definition of a stable network.

We denote the queue at junction i during time slot t as $Q_i(t)$. A network of queue backlogs is defined as strongly stable if:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Q_i(\tau)] < \infty, \forall i \quad (7)$$

Note that this performance resembles the qualitative performance of a single M/M/1 queue with arrival rate λ and service rate μ .

Furthermore, we denote as $f(\vec{x}(t))$ the penalty, which comes as an outcome of routing decisions between queues in time slot t . We assume that f is non-negative, continuous, convex and entry-wise non-decreasing, meaning, $f(\vec{x}) \leq f(\vec{y})$ when $\vec{x} \leq \vec{y}$ entry-wise. Let $f(\bar{x})$ be the penalty value of the function that works on the average value of the $\vec{x}$ vector. We formulate the road network as a stochastic optimization problem in which routing decisions minimize function f and keeping strongly stable queues simultaneously. That is:

$$\begin{aligned} &\text{minimise: } f(\bar{x}) \\ &\text{subject to: } \text{Strongly Stable} \end{aligned} \quad (8)$$

Assuming that $f(\bar{x})$ is a cost metric of the routing process, we can derive the solution of equation (8) using the Utility Optimal Lyapunov Networking framework [36, 37]. This can show that we may have the result of routing decisions resulting in a backpressure routing policy. More specifically, each junction computes the weight per outgoing road in every time-slot. The weight is given below:

$$w_{i,j} = (\Delta Q_{i,j} - \tau \theta_{i,j}) \bar{R}_{i,j} \quad (9)$$

where $\Delta Q_{i,j} = Q_i - Q_j$ is the queue backpressure and Q_i, Q_j are the backlogs of junctions i, j respectively, $\bar{R}$ is the road car driving rate and $\theta_{i,j}$ a road usage penalty that depends upon the particulars of the utility and penalty functions of (8). The parameter τ is a constant trades system queue occupancy for minimizing the penalty. We propose a decentralized approach where junction i calculates the backpressure weight of all its neighboring junctions. Thereafter, it is used to determine independent routing. For example, junction i locates the road (i, j^*) , which has the highest value of the backpressure weight as the next route of the car. We assume, at this point, that the weight is always larger or equal to 0. In the case that the weights are equal, we adjusted the algorithm to select the route with the shortest distance.

Figure 2. Backpressure Routing Operation

4 Penalty Minimisation

The main challenge we are facing is to identify the most suitable penalty function $f(\vec{x})$ in equation (8), in order to provide efficient performance when we use it in a real work road traffic network.

Initially, we give some preliminaries regarding the road traffic network problem. We model the road network using a directed graph $G(V, A)$, where V represents the junctions and A represents the roads. As we mentioned at a previous section, we assume that each junction has traffic control cameras, or other devices that may monitor and calculate the rate and driving time of each vehicle. The graph includes two attributes on each road $a \in A$. The first is an estimate before hand for its driving time for the solution we request, and the second is the road performance function. We use the road performance function that has been utilized in [31]. In particular, the road performance function, I_α , maps the traffic rate x_α , to its driving time $I_\alpha(x_\alpha)$.

The road performance functions, which will serve as the penalty functions of (8) determine the impedance of roads for a variety of congestion levels. We wish our functions to be nondecreasing and differentiable, as well as $I_\alpha(x_\alpha)$ to be convex. In our experiments we employ the function derived by the U.S Bureau of Public roads [26]

$$I_\alpha(x_\alpha) := I_\alpha^0 \left(1 + \alpha \left(\frac{x_\alpha}{c_\alpha} \right)^\beta \right) \quad (10)$$

where I_α^0 represents the travel time under no congestion, $\alpha \geq 0$ and $\beta \geq 0$ are tuning parameters, and finally, c_α is the practical capacity [38].

We employ the metric (10) to be minimized in the backpressure framework and we have the following penalty function in (8):

$$f(\vec{z}) = \sum_i \sum_{j \in N_i} z_{i,j}(t) \bar{I}_\alpha \quad (11)$$

where N_i is the set of neighboring junctions of junction i , I_α is the road performance and $z_{i,j}(t)$ is the number of routed cars over the road $i \rightarrow j$. Notably, $f(\vec{z})$ satisfies the properties of problem (8) and provides a backpressure weight, calculated by a junction i to a neighbor j as follows:

$$w_{i,j} = (\Delta Q_{i,j} - \tau \bar{I}_\alpha) \bar{x}_{i,j} \quad (12)$$

5 Byzantine Generals Algorithm for Electric Vehicle Fault Detection

The Byzantine General's problem is a consensus algorithm. In his seminal work Lamport et al. [18] proposed an algorithm of a scenario of a Byzantine General who decides on which strategy to follow, attack or retreat. To this end, the decision of the General has to be distributed to the lieutenants. However, there is a potential of traitorous behavior by the General of the lieutenants. Thus, the malicious officers will attempt to confuse the others by communicating a different action than the one they received.

In the EVRP problem as in [13], we denote the traffic light where the cars need to pass as the General. The lieutenants are the cars to be routed or the nearby traffic lights. We consider the cars and the traffic lights to communicate via a wireless system, in order for the next route to be determined. Hence, the traitors are thought to be the cars or traffic lights that exhibited some kind of malfunction in their systems.

In the Byzantine Generals algorithm the tests to be passed for consensus to be established are:

- A solution must guarantee that all the correct subsystems (cars, traffic lights) reach a decision on the value that they have been given.
- The correct subsystems need to conclude on the same value that they have been given.
- If the traffic light subsystem is a correct subsystem, then all the correct subsystems have to conclude to the value that has been given by the traffic light.

Of course, the major issue is the establishment of an agreement the source traffic light is a faulty process; thus communicating faulty values to the cars or nearby traffic lights. In our case, we can think of the source process to send a different backpressure weight to the other traffic lights/cars. This will result in different routing decisions for the same configuration regardless of the backpressure algorithm's decision. In figure 2 we show a faulty traffic light and the messages it conveys..

Lamports algorithm, which we employ in this work, assumes that there are n subsystems with m faulty subsystems where $n > 3m$. The traffic lights sends the messages. Thereafter, the other subsystems recursively forward the messages to all other subsystems. Finally we get the majority of the same messages that occur for the subsystems to follow.

Figure 3. Routes of Backpressure, Shortest Path and initial configuration **Figure 4.** Backpressure Routing Rounds **Figure 5.** Faulty Traffic Light

6 Simulations

We produced a road network in MATLAB consisting of 7 junctions that we assume they are traffic lights equipped with traffic cameras. Furthermore, we have 10 roads connecting the junctions in a fashion that is given in figure 3 (a). We also show the distance of the roads between the junctions. This

is since we wish to compare our algorithm with Dijkstra's shortest path algorithm [29].

The parameters that the backpressure algorithm uses are given in table 2. More specifically, we provide the times that a car requires to pass from a road and the queue backpressure for each junction. Moreover, the rate of each road is assumed to be equal, notably 4 cars per round (traffic light green to red). Finally, we set the parameter $\tau = 2$. We assume that there are certain roads that require more time to be driven even though they are of longer distance, due to more cars selecting these routes or by an occurrence of accidents.

Table 1. Time and Queue Backpressure per Junction

Road	Time	Queues (i,j)	Queue Backpressure
A-B	4	15-10	5
A-C	6	12-5	7
A-D	10	13-6	7
B-C	10	10-5	5
B-F	7	10-3	7
C-E	5	5-2	3
C-F	5	5-3	2
F-S	4	3-1	2
E-S	5	2-1	1
D-S	12	6-1	5

Initially, we run the algorithm computing the shortest path for a car starting from junction A to the destination S. We observe that the route computed is $A \rightarrow C \rightarrow E \rightarrow S$. This can be easily seen from the distances provides for each road.

Figure 6. Byzantine Generals Operation 8 cars/traffic lights

Figure 7. Byzantine Generals Fault Detection Scenario

Thereafter, we applied the backpressure algorithm, where we see that the route selected by a car starting from the origin A , in order to reach the destination S is different from the shortest path algorithm. The car drives to junctions $A \rightarrow C \rightarrow F \rightarrow S$. Furthermore, the driving time of the route using the backpressure as opposed to the shortest path algorithm is shorter by 1 minute. We are positive that in other scenarios the time difference may be longer.

In figure 4 (a) - (c) we present the steps of the backpressure algorithm by showing the routes selected by cars in different junctions. Each vertex includes the letter of the junction and its respective queue size i.e., $A - 14$. Furthermore, we provide the backpressure weight in each edge of the graphs, in order to make the route selections clear. Finally, we converted the driving time from minutes to hours to calculate the backpressure weights. Note that we assume that the incoming cars to junction $A = 3$ and that the junctions' traffic lights work simultaneously, for the sake of simplicity.

During the first round that the traffic lights become green we have the following configuration, which appears in figure 4 (a). As we can see, the backpressure weights that are given in the edges of the figure, promote the cars to route from $A - C$, $B - C$, $C - E$, $D - S$ and $F - S$. Note that the routes to S occur, since there are no other routes from the respective junctions. Furthermore, we assume that cars routed to junction C from junction A arrive first, since the driving time of the road is less than than the road originated from junction B .

In round 2 of the backpressure algorithm in figure 4 (b), we observe that the car that interests us moves from junction C to junction F , since the backpressure weight is higher than the one from C to E . Furthermore, the other routing decisions in this round consist of $B - F$, $A - D$ and F, E, D to the destination S , since there are no other routes to select from. Moreover, we observe that these routes have a weight of 0, since it is the smallest value our algorithm may accept (the weight has a negative value in our calculations).

In figure 4 (c), the car that we are routing, reaches the destination S from junction F . We also see that other routing decisions include $A - C$, $B - C$, $C - E$ and all the one hop junctions to the destination. Note that the routing decision for the routing of the cars of junction C to either F or E has been done using the shortest path, since both the respective backpressure weights are 0. The end of this round routes our car from junction A to the destination S .

6.1 Fault Detection Results

We assume that we have 8 cars/traffic lights that make their routing decisions based on backpressure routing. In our first scenario, we have two faulty cars/traffic lights that will have to be detected. In the case that there is a faulty process sending different messages to the other cars/traffic lights, the cars will be routed by forming a consensus of their communication with the other cars/traffic lights. The General is process

3.

In Table 2 we see the consensus reached after the operation of the Byzantine Generals algorithm. The cars/traffic lights communicate between them via their wireless systems.

Table 2. Output of Byzantine Generals Operation 8 cars/traffic lights

Road Number	Round 1
0	1
1	1
2 (General)	faulty
3	faulty
4	1
5	1
6	1
7	1

Even though our General traffic light is traitorous, we observe that the remaining cars/traffic lights have come to an agreement reaching the value 1. Thereafter, we show the message tree of car/traffic light 4, in order to show the steps of the final decision. Note that car/traffic light 2 is also faulty; however, that does not seem to affect the consensus of the cars/traffic lights.

In Figure 6, we analyzed the messages communicated involving car/traffic light 4. The source subsystem, which is faulty, provides different values in its messages to the other subsystems. Process (subsystem) 4, then gets the values from the other subsystems and in its neighboring table, it clarifies that it gets a number of faulty messages. However, in the final round the messages it gets from the other subsystems participating in the consensus process, indicate an agreement, since there are 4 processes that result in the value 1. Hence an agreement is achieved, bypassing the faulty values by the source and subsystem 3 (process 2). In this way the subsystem, is able to determine a routing strategy that will rule out possible flaws in the communication. In the same way, the rest of the processes reach the same value.

Thereafter, we experimented we the byzantine generals algorithm using the configuration where 5 cars/traffic lights where the source process is faulty.

We can see in table 3 the outcome of the algorithm where the source is faulty providing different evaluations of the roads. The remaining traffic lights/cars result to the same evaluation of the roads; thus making the correct decision regardless of the faulty traffic light/car transmission. We show the faulty source traffic light/car in figure 7 (a). We can see the difference in the incoming values from the peer traffic light/car and the outgoing values to the remaining traffic lights cars. The algorithm continues further until it reaches a consensus.

In Figure 7 (b) we see the messages exchanged by the source traffic light/car which is faulty. The message sent to the other processes is marked by X, since the process is faulty. Hence the outgoing messages are uncertain. Hence, when the algorithm continues the consensus of the remaining traffic lights cars will be reached by selecting the value 1.

Table 3. Output of Byzantine Generals Operation 5 cars/traffic lights

Road Number	Round 1
0	1
1	1
2 (General)	faulty
3	1
4	1

7 Conclusions

In this paper we applied a state-of-the-art routing protocol for wireless networks to electric vehicle routing. The backpressure routing is essentially routing without routes. It calculates the next route of the car dynamically based on the calculation of the backpressure weight and the strong stability of the formed queues on the junctions. We utilized the driving time of each road as a penalty to be minimized by every car. The penalty is selected based on the necessity of a company's fleet to reach the clients' addresses within a given time window and in a single depot.

Our simulations showed that the backpressure algorithm creates faster routes than Dijkstra's shortest path algorithm. This may be very useful, since roads may have several problems such as traffic accidents, road works or traffic jams, which may prolong their driving times even if the distance between two junctions is small. Hence the backpressure algorithm establishes an efficient routing mechanism.

Our future work includes the integration of the electric vehicle battery to the penalty minimization process. This might provide a useful insight as to the route selection problem, since the fastest route may not be the most energy-efficient one. It is crucial to compare our approach with state-of-the-art routing protocols for EVs that take into consideration charging station locations and charging times. The backpressure framework provides us with the flexibility to employ the penalty function of our liking, in order to produce the backpressure weight. Hence we are optimistic that it will perform well comparing other routing schemes. Moreover, our approach is not restricted to EVs only and it may be adjusted to operate in conventional vehicle routing as well. Finally, we aim to put the backpressure algorithm to a more complex road network, and perhaps to a real test.

Moreover, we setup two test scenarios of 8 and 5 cars/traffic lights respectively to investigate the application of the Byzantine Generals algorithm for fault detection. In the backpressure routing decision. In our first scenario, we have planted two faulty cars traffic lights among the 8 processes and we executed the algorithm to detect the faulty processes. The faulty processes have been detected successfully, meaning that a wrong routing decision due to communication malfunction has been prevented. Similarly, in our second scenario we shows the consensus reached when the General car/traffic light is malfunctioning, and we showed the message trees of a healthy and faulty car/traffic light.

Our future work includes the integration of the Byzantine Generals algorithm to the Backpressure routing protocol as a fault prevention mechanism. Furthermore, we aim to put the

two methodologies to a real test, due to the effectiveness of the Byzantine Generals algorithm and the simplicity of the Backpressure routing protocol.

Acknowledgements

Special thanks to Professor Alexander Wolf from the Department of Computing of Imperial College London for his seminal lecture on the Byzantine Generals problem.

REFERENCES

- [1] A. Emadi, Transportation 2.0, IEEE Power and Energy Magazine, vol.9, no.4, pp. 18–29, 2011.
- [2] A.Artmeier, J.Haselmayr, M.Leucker, M.Sachenbacher *et al.*, The optimal routing problem in the context of battery-powered electric vehicles, in *Workshop: CROCS at CPAIOR-10, Second International Workshop on Constraint Reasoning and Optimization for Computational Sustainability, Bologna, Italy*, 2010.
- [3] N.Touati-Moungla and V.Jost, Combinatorial optimization for electric vehicles management, *Journal of Energy and Power Engineering*, vol.6, no.5, 2012.
- [4] Y.Lim, H.-M. Kim, and S.Kang, Information system for electric vehicle in wireless sensor networks, in *Communication and Networking*. Springer, 2010, pp. 199–206.
- [5] S.Moschoyiannis, L.Maglaras, J.Jiang, F.Topalis, and A.Maglaras, Dynamic wireless charging of electric vehicles on the move with mobile energy disseminators, *Int'l Journal of Advanced Computer Science and Applications (IJACSA)*, vol.6, no.6, pp. 239–251, 2015.
- [6] M.Gerla, E.-K. Lee, G.Pau, and U.Lee, Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds, in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 241–246.
- [7] J.M. Anderson, K.Nidhi, K.D. Stanley, P.Sorensen, C.Samaras, and O.A. Oluwatola, Autonomous vehicle technology: A guide for policymakers. Rand Corporation, 2014.
- [8] E.D. Spyrou and D.K. Mitrakos, A backpressure framework applied to electric vehicle routing, *BMSD'16, Rhodes*, 2016.
- [9] P.-E. Dumont, A.Aitouche, and M.Bayart, Fault detection of actuator faults for electric vehicle, in *2007 IEEE International Conference on Control Applications*. IEEE, 2007, pp. 1067–1072.
- [10] M.A. Djeziri, R.Merzouki, B.O. Bouamama, and M.Ouladsine, Fault diagnosis and fault-tolerant control of an electric vehicle overactuated, *IEEE transactions on vehicular technology*, vol.62, no.3, pp. 986–994, 2013.
- [11] C.Sankavaram, B.Pattipati, K.Pattipati, Y.Zhang, M.Howell, and M.Salman, Data-driven fault diagnosis in a hybrid electric vehicle regenerative braking system, in *Aerospace Conference, 2012 IEEE*. IEEE, 2012, pp. 1–11.
- [12] G.Kiokos, E.Zountouridou, C.Papadimitriou, A.Dimeas, and N.Hatziaargyriou, Development of an integrated wireless communication system for connecting electric vehicles to the power grid, in *Smart Electric Distribution Systems and Technologies (EDST), 2015 International Symposium on*. IEEE, 2015, pp. 296–301.
- [13] E.D. Spyrou and D. K. Mitrakos, Wireless Communication Fault Detection in the Electric Vehicle Routing Problem IEEE CoDIT, 2017, to appear
- [14] M.A. Eshraghi, M.S. Sariri, V.Schwarzer, R.Ghorbani, D.Block, and F.S.E. Center, Islanding detection and over voltage mitigation using wireless sensor networks and electric vehicle charging stations, 2016.
- [15] J.Wan, H.Yan, D.Li, K.Zhou, and L.Zeng, Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle *The Computer Journal*, vol.56, no.8, pp. 947–956, 2013.
- [16] M.Neely Notes on backpressure routing *Course of Stochastic Network Optimization*, 2011.
- [17] S.Moeller, A.Sridharan, B.Krishnamachari, and O.Gnawali, "Routing without routes: The backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010, pp. 279–290.
- [18] L.Lamport, R.Shostak, and M.Pease The byzantine generals problem *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol.4, no.3, pp. 382–401, 1982.
- [19] Abousleiman, R. and Rawashdeh, O. (2014). Energy efficient routing for electric vehicles using particle swarm optimization. Technical report, SAE Technical Paper.
- [20] Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., and Margaritis, D. (2014). Electric vehicle routing problem with industry constraints: trends and insights for future research. *Transportation Research Procedia*, 3:452–459.
- [21] Artmeier, A., Haselmayr, J., Leucker, M., and Sachenbacher, M. (2010). The optimal routing problem in the context of battery-powered electric vehicles. In *Workshop CROCS at CPAIOR-10, 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*.
- [22] Baheti, R. and Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12:161–166.
- [23] Baum, M., Dibbelt, J., Gamsa, A., and Wagner, D. (2014). Towards route planning algorithms for electric vehicles with realistic constraints. *Computer Science-Research and Development*, pages 1–5.
- [24] Bondy, J.A. and Murty, U. S.R. (1976). *Graph theory with applications*, volume 290. Macmillan London.
- [25] Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221–228.
- [26] BUREAU, O. P.R. (1964). Traffic assignment manual. *US Department of Commerce*.

- [27] Dantzig, G.B. and Ramser, J.H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.
- [28] deWeerd, M.M., Stein, S., Gerding, E.H., Robu, V., and Jennings, N.R. (2015). Intention-aware routing of electric vehicles.
- [29] Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- [30] Emadi, A. (2011). Transportation 2.0. *Power and Energy Magazine, IEEE*, 9(4):18–29.
- [31] Jahn, O., Möhring, R.H., Schulz, A.S., and Stier-Moses, N.E. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4):600–616.
- [32] Kopetz, H. (2011). Internet of things. In *Real-time systems*, pages 307–323. Springer.
- [33] Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- [34] Moeller, S., Sridharan, A., Krishnamachari, B., and Gnawali, O. (2010). Routing without routes: the backpressure collection protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 279–290. ACM.
- [35] Nagata, Y., Bräysy, O., and Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737.
- [36] Neely, M.J., Modiano, E., and Li, C.-P. (2008). Fairness and optimal stochastic control for heterogeneous networks. *Networking, IEEE/ACM Transactions on*, 16(2):396–409.
- [37] Neely, M.J. and Ugaonkar, R. (2008). Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2152–2158. IEEE.
- [38] Patriksson, P. (1994). *The traffic assignment problem: models and methods*.
- [39] Little, John DC, and Stephen C. Graves. Little’s law. Building intuition. Springer US, 2008. 81-100.
- [40] Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.
- [41] Touati-Moungla, N. and Jost, V. (2012). Combinatorial optimization for electric vehicles management. *Journal of Energy and Power Engineering*, 6(5).
- [42] Neely, Michael J., Eytan Modiano, and Charles E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications* 23.1 (2005): 89-103.
- [43] Felipe, ngel, et al. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review* 71 (2014): 111-128.
- [44] Lin, Jane, Wei Zhou, and Ouri Wolfson. Electric vehicle routing problem. *Transportation Research Procedia* 12 (2016): 508-521.
- [45] Adler, Jonathan D., et al. The electric vehicle shortest-walk problem with battery exchanges. *Networks and Spatial Economics* 16.1 (2016): 155-173.
- [46] Chen, Jinbo, Mingyao Qi, and Lixin Miao. The Electric Vehicle Routing Problem with Time Windows and Battery Swapping Stations. *Industrial Engineering and Engineering Management (IEEM)*, 2016 IEEE International Conference on. IEEE, 2016.
- [47] Hung, Ying-Chao, and George Michailidis. Optimal routing for electric vehicle service systems. *European Journal of Operational Research* 247.2 (2015): 515-524.