A SUMO Based Evaluation of Road Incidents' Impact on Traffic Congestion Level in Smart Cities

David Smith, Soufiene Djahel, and John Murphy
Lero, School of Computer Science and Informatics, University College Dublin, Ireland
David.Smith.1@ucdconnect.ie,{soufiene.djahel, j.murphy}@ucd.ie

***Abstract*—Recently, the increasing road traffic congestion has attracted a lot of attention from the research community aiming at proposing innovative solutions to reduce the huge economic loss incurred by this problem. In this paper, we first evaluate the impact of random road incidents on the commuters travel time and the overall traffic congestion level under several scenarios, and provide comprehensive analysis of the obtained evaluation results. Then, we propose an extension of the open source traffic simulator SUMO (Simulation of Urban MObility) to enable real-time vehicles re-routing, to bypass the blocked road due to an incident, by updating their predefined static routes during simulation runtime. The proposed re-routing mechanism has been implemented and the obtained results have proven its high efficiency on reducing the commuters travel time, in case of accident, compared to the basic SUMO.**

***Keywords* – ITS, Smart Transportation, Vehicles Re-Routing, SUMO, Road Incidents, Smart Cities.**

I. Introduction

In big cities, road traffic congestion is caused by the volume of traffic closely approaching the maximum capacity of the road network. During rush hours, it gets worse and with more people joining the road network every day, the congestion problem will not disappear on its own. It is infeasible for a government to match a road network improvement programme to the unrestricted trends in traffic growth.

The economic loss due to traffic congestion is a huge factor, in the 39 metropolitan areas of the US with a population of 1 million or more, roughly one third of all vehicular travel occurs under congested conditions where the average speed is half of its free flow value. It is stated that half of the congested traffic occurs in express ways, causing a delay of over half a minute per kilometre of travel. The other half is on other arterial roads where the delay amounts to 1.2 minutes per kilometre of travel. With some 75 million licensed drivers in heavily populated areas, each averaging roughly 16,000 kilometres per year in those areas, there are 1.2 trillion kilometres driven in metropolitan areas, this amounts to a total delay of 6 billion hours [1]. Back here in Europe the economic cost is an even bigger factor, with estimates of 200 billion Euros in losses due to traffic congestion in 2012. This is roughly 2% of Europe's GDP, and is more than double of the American estimate of $101 billion [2]. Therefore, it is clear from this data that there is a real necessity to provide efficient tools to alleviate the impact of the traffic congestion. To this end, navigation system companies have designed some tools but they are still in the early stages of becoming fully fledged and highly reliable products. To develop a more sophisticated system, simulation tools must be used to gain more insights on the traffic flow patterns on road networks and how this traffic flow evolves if, for example, an accident occurs or road works are taking place etc.

The most used navigation systems for drivers and pedestrians are certainly TomTom NV [5] developed by a Dutch manufacturer in Europe, and Garmin [6] developed in USA. Both systems use GPS to determine the location and some routing algorithms to establish (generally) the fastest route to the destination. In some cases, we can also choose routes that avoid toll roads and motorways. TomTom has recently developed a system, known as TomTom Traffic, which provides the users with accurate real time information based on the state of the traffic and any congested routes ahead. It pinpoints exactly where the expected delays start and end, and if the traffic situation changes then TomTom will continuously look for the fastest route. It, therefore, significantly enhances the users awareness, by informing them where incidents have been occurred and giving them real time congestion information, so that they can decide whether to stay on course or take a detour.

Garmin offers a similar live service that updates every two minutes to check traffic situations. Data is pulled from millions of other users, including mobile phone users, incident reports, radio feeds, news feeds, historical traffic data and fixed traffic sensors. These systems have only recently been introduced in Ireland, with TomTom Traffic only going live in 2011, and Garmin in 2010. Although these systems pull data from a lot of users, tens of thousands in Ireland, it is broadly limited compared to the amount of actual road users. It also uses the number plate registration systems on motorways. This is a fixed infrastructure installed by local authorities that tracks a number plate and how long it takes to get from one point to another. Again these systems are only installed on major motorways, and do not take into account arterial roads approaching or exiting these motorways. It is clear that although these systems are currently in use by millions of people around the world, their scope is rather limited when it comes to specific areas not monitored by sensors.

The main aim of this paper is to extend the microscopic traffic simulator (SUMO) by designing a re-routing mechanism that ensures real time update of the drivers routes (i.e. dynamic route update) upon detection of any abnormal increase in traffic congestion or as consequence of an accident. Our ultimate goal is to develop a model that will update a vehicle's route when an accident occurs. The updated route will be based on the shortest path from the vehicle's location (upon detection of an accident), to its destination, excluding the road where the accident occurs.

The remainder of this paper is organized as follows. In section II, we present the literature followed by a detailed discussion of some missing features in the current release of SUMO in section III. Section IV evaluates the impact of accidents on road traffic congestion and commuters travel time to their destinations, while section V describes the proposed re-routing mechanism to alleviate the above impact and presents the obtained simulation results. Finally, section VI concludes the paper.

II. Related Works

Simulation of **U**rban **MO**bility, (SUMO) [13], is an open source, microscopic, multi modal traffic simulator. It allows the user to simulate how a specified traffic demand performs on a given road network. It is microscopic, which means each vehicle is modelled explicitly, it has its own route and moves individually through the network. The German Aerospace Center began developing SUMO in 2001 and since then it has been improved and has evolved into a suite of traffic modelling utilities which includes a road network capable of reading different source formats, demand generation and routing utilities. SUMO was developed as an open source simulator aiming that its prospective users will suggest and implement improvements to the simulator helping to build a better and more realistic model. SUMO is not a traffic simulator only, but a suite of applications that allow the user to create/import a road network and define its corresponding traffic demand. It uses "netconvert" to import a network from Open Street Map or from other traffic simulators such as VISUM, MATsim or VISSIM. Once a road network is imported and converted to the appropriate format, traffic demand, and routes for each vehicle should be created. "*DUAROUTER*" is one of the tools used to compute routes, "*DFROUTER*" is another. These routing tools take the network and trips as arguments and produce a route file that contains the routing information for each vehicle defined in the network. We can use random trips (which is what has been used in this paper), or we can manually create a demand using OD (Origin-Destination) matrices or even by supplying various parameters for the specified network. These parameters include the population and the land class usage definition, among others.

In a research paper by Vi Tran Ngoc Nha *et al* [16], the various routing algorithms that could be used in conjunction with SUMO or another traffic simulator were described and deeply analysed based on their merits and limitations. The first of these algorithms that would be a suitable candidate of dynamic routing is Dijkstra [19]. This algorithm finds the shortest path with the lowest cost from one node to all the nodes in a city map. Dijkstra is a worthwhile algorithm because it terminates once the destination node is found, i.e. shortest path found. Some other algorithms are unable to determine the shortest path until all the nodes are formed into a shortest path tree. An alternative to this would be A* [17] which uses a heuristic function instead of an optimal search algorithm. Therefore, A* is able to restrict the search space which in turn improves computation time, the search space would be reduced to the area where an incident has occurred, such as an accident or sheer traffic volume. The authors then discuss how these route planning algorithms can be improved. For example, they highlighted some parameters that can be used as inputs for the routing algorithm, such as road information, the current state of traffic volumes and congestion on the road, the destination location, in addition to other parameters such as the fuel consumption level, the vehicle driving conditions and the driver readiness for driving (i.e. if he is tired and needs a break etc). The best route selection criteria and algorithm evaluation metrics are discussed, to identify the most appropriate route parameters to consider, such as travel distance, travel time, the ease of driving and the travel cost. A simulation based comparison of the above routing algorithms has been performed in [18] wherein the authors presented extensive simulation results using a real traffic data from TAPASCologne project.

INRIX, Inc.[20] is a provider of traffic information, it provides historical and real-time traffic information throughout the US and Canada as well as most of Europe and Brazil. INRIX collects information about vehicles speeds from almost 100 million anonymous mobile phones, delivery vehicles and lorries, along with other various fleet vehicles that have been equipped with a GPS device. All data collected is processed in real time and used to create traffic flow information for motorways and arterials across its user space. As it was mentioned in section I, TomTom and Garmin have developed systems that alert drivers of any congestion ahead before they reach it. Both TomTom and Garmin have a partnership with INRIX as a part of their data collection process; they also use many other sources as was mentioned above. They are both commercial services that cost quite a lot and considering they only work proficiently in big cities with a lot of user data and fixed infrastructure input, they might not be suitable for users driving in sub-urban or rural areas.

The usefulness of these services is highly reliant on their reaction times. They cannot always avoid congestion, since this latter can often be spontaneous due to an accident. Google Maps and Microsoft's Bing use statistical predictive analysis that estimates where congestion may begin and end, but due to the volatile and unpredictable nature of congestion and accidents, these tools are only useful to a certain extent. These predictive systems, which TomTom and Garmin also employ, rely on the recurring congestion trends. This only accounts for 50% of all congestion [21], so in order to assess the real impact of congestion, caused by random incidents, on road traffic simulation tools must be used. Therefore, the system which we will build in this work will simulate random incidents and test the performance of our proposed re-routing mechanism.

III. Missing Features in SUMO

On the SUMO website they have a ticket system, whereby the users can add tickets which are jobs that need to be done, sometimes something as simple as bugs or typos, but amongst the bugs are suggestions for more substantial improvements. They also include a student and support page with possible suggested projects. Among the list of projects are topics related to traffic science, information science and other issues. The suggestions related to Traffic Science include Pollutant Emission modelling and Evaluation, which would involve modelling the amount of emissions from given road networks based on a realistic traffic flow throughout an average day or week. Another suggestion was a traffic light comparative study involving testing various traffic light algorithms performance for certain traffic loads.

Regarding the route choice and demand modelling there is a lot more suggestions. The evaluation of one-shot traffic assignment function that assigns a route to each vehicle on the network at the start of simulation. Another route choice suggestion is to design an alternative method for shortest path search in large networks to reduce the high computation cost of Dijkstra. Other suggestions consist in exploring the use of induction loop values for Highway demand generation or to extrapolate routes based on these values. Moreover, forecasting of Demand Time-Lines was also suggested. This would involve running simulations on real road networks and applying realistic traffic flow to find out the time periods during which some road segments become highly congested.

In traffic simulation models section, some missing features have been highlighted such as, the need for concrete validation of simulation (e.g. Tapas Cologne work [23], as well as the development of simulation model to simulate emergency service vehicles. This would be a very useful simulation, since during rush hours, accidents often occur, and with road networks at their maximum capacity in most cases, it is very difficult for emergency services to reach their destination quickly. Therefore, simulation of hundreds of destinations on a road network from a hospital or police station will be a great source of information for creating a specialized GPS system for emergency vehicles. Another missing feature which would greatly affect traffic congestion alleviation in SUMO is making use of the opposite lane for overtaking, since in SUMOs latest release a vehicle is stuck on its side of the road, if a vehicle gets stopped behind a stalled or crashed car then it is essentially trapped, whereas in reality it would just overtake and keep going. In some cases, in SUMO, a car can overtake a crashed vehicle if the road they are on has more than one lane going in the same direction (e.g. dual carriageway). Moreover, the current release of SUMO doesn't support left hand driving scenarios, which limit the applicability of this simulator to right hand driving scenarios only.

Finally, under SUMO simulation models section, it is mentioned that pedestrian flow is one of the future models to be included into the simulator since during rush hours, pedestrian traffic has an important impact, especially in and around train station and bus stops, where footfall would be very high, and traffic lights have to alternate more often thus slowing down traffic. Therefore, introducing such model will enable more accurate traffic simulation. One of the major missing features that were found while getting to understand SUMO is how the route definitions are all made prior to runtime by default. In order to update these routes we have to add additional components using TraCI [14] alongside SUMO so that the routes can be redefined during runtime. According to the study in [15], SUMO should have more than just the shortest path to calculate the route, however, its current release only takes into account the shortest path from the source node to the destination node.

IV. Investigating the Impact of Accidents on Road Traffic Congestion

The simulation tool we used to simulate a road network and demand on that network is SUMO [13]. This simulator allowed us to generate traffic demand on a small grid network (using NETGENERATE) to begin with, enabling us to see what was happening during runtime. This was an essential step because OpenStreetMaps are often too complex to see exactly what is happening. This small scale map was used to build our initial experiments on, by first creating an accident.

A. How to create an incident?

An incident refers to anything that stops the flow of traffic. Examples in the real world could be a collision, road works, bad weather causing very slow speeds, or just sheer volume of traffic. There are a few options in SUMO to simulate such an event. Stopping a car for a fixed period of time, by defining a point along its route when it should halt and for how long. Manipulating traffic lights so that one stays red for a longer period of time than it should. Or setting the speed limit on an edge so low that the traffic is practically not moving. We decided to employ the first option, since stopping a car was the easiest of the three to implement and therefore easy to port from one scenario to another. In order to stop a car on a given edge we first need to get the edge and lane IDs. Then, we look up the edge list and include the following code in the vehicle definition that we want to stop to simulate a traffic jam.

$$< stoplane = "0/0to1/0\_0" endPos = "10" duration = "200" / >$$

This code stops the vehicle on the specified lane at 10 meters for 200 seconds duration. Figure 1 shows that the car at the bottom of the screen is being halted, which in turn stops all cars that are behind it or trying to get onto the route. Various problems arose when it came to initially getting a car to stop. Figuring out how long to halt for a realistic traffic accident. From driving experience on the road (not in accidents) we figured ten to twenty minutes would be sufficient to impact on the traffic on the route, as well as being a realistic time in which the vehicles involved are left blocking a road, obviously more severe accidents can take hours to clear but in reality these more serious accidents tend not to happen in low speed urban environments.

Another issue we have faced when stopping a vehicle was teleporting. In SUMO, when a vehicle waits for more than 300 seconds it teleports by default. This means that when a car is stopped, to simulate the occurrence of an accident, for a desired realistic time of 1200 seconds, the cars behind the crashed car will be stopped during the first 300 seconds only and then will teleport. In SUMO, a car teleports by

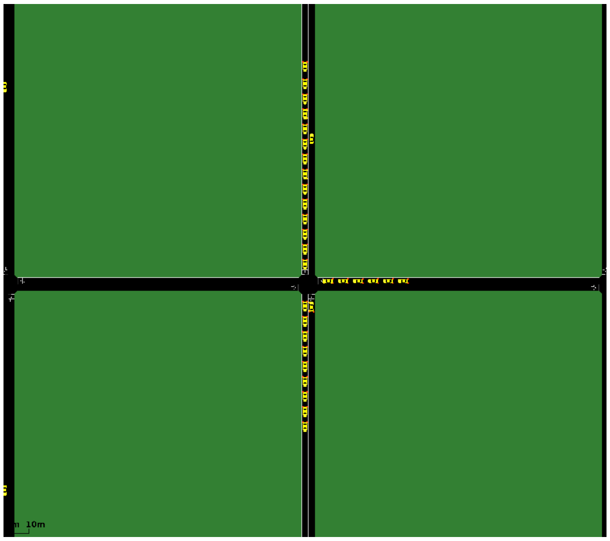

Figure 1: Car causing a traffic jam

going "under" the road, and popping back up at the first chance it gets when the road ahead is clear. In order to overcome this issue, the time to teleport was increased to 1200 seconds, which led to yet another issue. We discovered that teleporting was a bug prevention measure to tackle random collisions, as in SUMO collisions aren't technically defined but are observed. To overcome them, the two or more vehicles involved in a collision are teleported. Since the time to teleport was increased to establish more realistic travel times, any collision that occurred further increased traffic jams and caused congestion that could not be prevented. These collisions were mainly due to traffic light issues and road networks not being properly converted using SUMO's NETCONVERT tool (used to convert OpenStreetMaps to SUMO networks), as it doesn't support left hand driving maps yet. Therefore, we decided to use more straight and grid-like networks of US cities as shown in Figure 2, instead of Dublin.

Other arguments can be made for not using SUMO to simulate Irish or left-hand road networks in general. The main reason is that SUMO is by default a right-hand road simulator. When a left hand road network is fed into NETCONVERT it gets most of the connections right. However, it cannot handle roundabouts, instead of inverting the direction it remains as it is and sends the traffic the left way around instead of right. It also cannot handle one way streets or dual-carriageways or motorways correctly, since these roads have explicit directionality, it goes with whatever this definition is and sends the traffic down the road the wrong way. This might seem trivial but when we have more than one lane of traffic trying to essentially swap sides of the road collisions will occur.

B. Evaluation results

When the simulation was run without any manually inserted incidents, a fixed travel time would be generated in SUMO. To output this data in the form of a file some minor configurations had to be made. Within the output file it lists various metrics of the route, departure time, arrival time, trip duration etc. this is in XML, which needed to be parsed. Initially the simulation was run with no accident and the trip duration data was written to a file, then an accident was introduced and that new trip duration was written to a separate file. In order to compare the two results a HashMap was required, using the VehicleID as the key. This way the no-accident data could remain the same whilst various additional tests could be run, such as modifying the number of accidents. Only the vehicles that returned an increase in travel time of more than 2 minutes were considered, this was done to reduce the error rate. Some random vehicles would have a reduced travel time and other vehicles had a minimal change of less than 2 minutes, while some other vehicles had seemingly ridiculous times, so any times above 4000 seconds were discarded. These were chosen as cut off points because on the lower end it is 10% of the accident time, and 4000 seconds sets the upper bound high enough to eliminate outliers, which proved to be a more realistic representation of what could be seen on the network when running the simulation. These bounds returned a subset of the most affected vehicles, and these were the ones taken into account for the results.

The histogram depicted in Figure 3 illustrates the impact of the number of accidents on the travel time of vehicles. The times shown are the average increase in travel time with respect to the trip duration when no accident occurs; this value is calculated as shown in the formula below.

$$AverageIncrease = \frac{\sum_{i=1}^{n} TT_{increase}(vi)}{n} \tag{1}$$

where

$$TT_{increase}(vi) = TT_{afterAccident}(vi) - TT_{beforeAccident}(vi)$$

and TT = Travel Time, vi = every affected vehicle, n = number of vehicles

In the three scenarios shown in Figure 3, the Grid Network is a 10x10 Grid similar to the one shown in Figure 1, Los Angeles (LA) is an extract of the area in and around Hollywood and New York (NY) is the area of Lower Manhattan. The Grid Network has such consistent results because there are no traffic lights just crossroads LA was more volatile because of the nature of the topology; it varies from very wide multi-lane roads to small single lane each way traffic. New York like the Grid was also consistent since New York is essentially a Grid, but since it has more restrictive movement with lots of traffic lights the travel time increases as the number of accidents increases. Figure 4 illustrates the ratio between the average increase in travel time and the average trip time without accident. The Grid Network had very short trip durations without accident; this meant the 1200 second accident had a greater percentage increase than the other two scenarios. New York's' average travel time was almost half an hour, so a 20 minute delay did not have as much of an effect

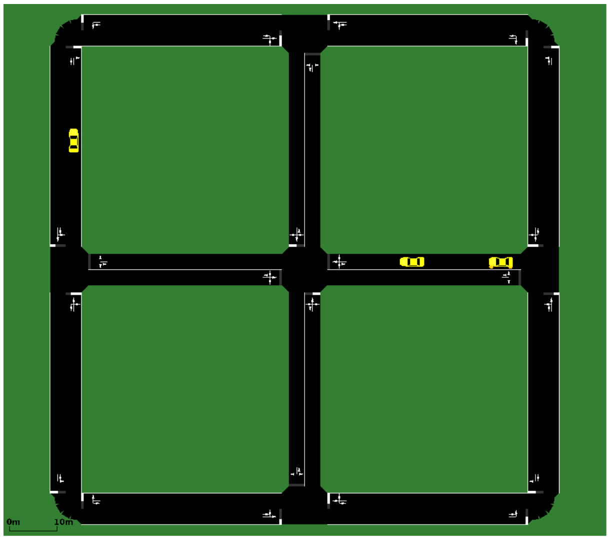

Figure 2: Simple 3x3 Grid Network

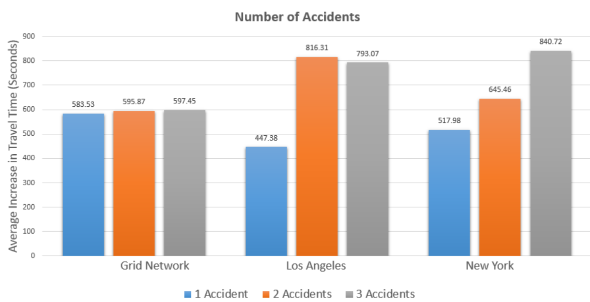

Figure 3: Impact of the number of occurred accidents on the increase in travel time

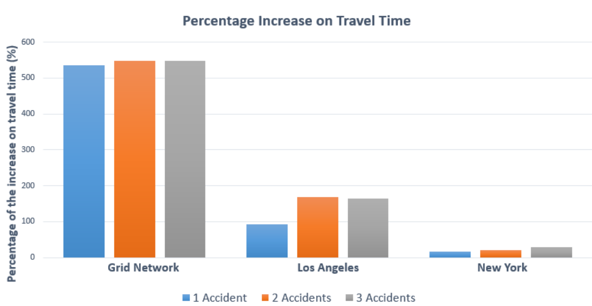

Figure 4: Ratio between increased travel time and average travel time without an accident

on the overall average increase. Notice that all of the above results have fixed vehicle density between tests, equals to 3600 cars per hour, as well as fixed accident duration equals to 20 minutes. The location and timing of accidents are the main metrics which we have varied during simulation in order to highlight their impact. We will illustrate with the grid network shown in Figure 5 how these factors affect the travel time.

Figure 5 illustrates how the location and timing of the

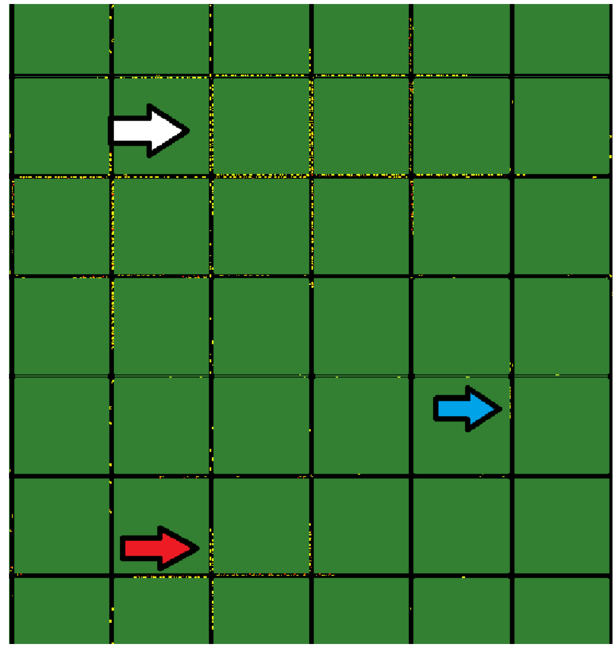

Figure 5: Order of Accidents: White - 1st, Red - 2nd, Blue - 3rd

accidents affect the overall increase in travel time. The biggest jam is at the 1st accident location (i.e. white arrow, notice that every yellow dot represents a car), therefore less traffic has the opportunity to get to the other accidents locations. This explains the lack of a major jump in travel time with respect to the number of accidents occurring in the Grids network results discussed above. Another experiment t was carried out to highlight the impact of road traffic density level on the overall increase in travel time. For each scenario shown in Figure 6, the accidents data was averaged and used as an overall average increase in travel time with respect to the traffic density. The vehicles density was then doubled to 480 cars per km^2 and the same tests were run. From the plotted histogram we see that the doubled traffic density has a similar impact across all scenarios. The Grid and NY networks had the highest increase in travel time as they have short streets with a lot of traffic lights and junctions, whereas for LA network the impact was lower due to the greater number of lanes on each road and the less frequent traffic lights.

As we can see from the results discussed above, the network topology is another important factor that affects the increase in travel time. Figure 7 illustrates this effect by aggregating all simulation results from previous tests. As expected, the grid and New York networks are the two which were most adversely affected by their topology, since they both have a lot of single lane roads and in NY there are mostly one way streets, in the contrary of LA network which, as mentioned above, characterized by very wide streets and very few one way streets.

The last comparison that we have conducted was the evaluation of the accidents duration impact on the overall increase

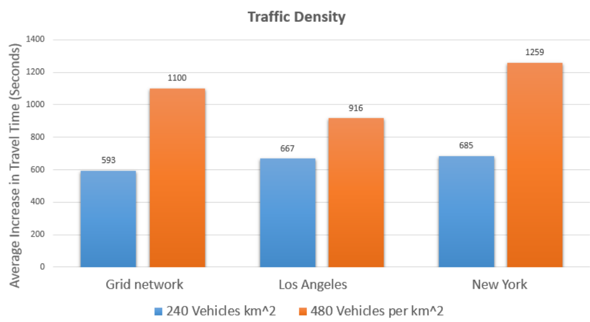

Figure 6: Impact of traffic density on Travel Time

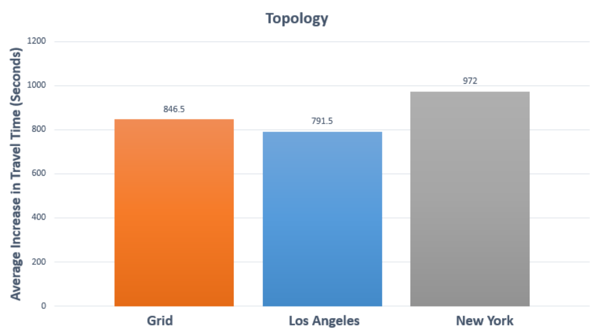

Figure 7: Impact of topology on the increase in travel time

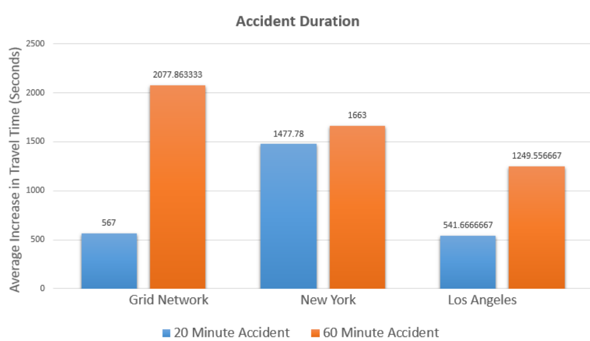

Figure 8: Impact of accident duration on travel time

in travel time. In order to carry out this test the duration of the accident was tripled, to 60 minutes. This data was collected by averaging the increase in travel time for three accidents for both durations (i.e. 20 minutes duration and 60 minutes duration). From the results plotted in Figure 8, it is clear that the accident duration further increases the travel time, but the topology of the networks really comes into account here. The Grid Network became jammed, almost entirely when an accident lasts for so long. New York network, as it is similar to the Grid network, had the same problem, but since there was such gridlock on the network to begin with, the increase in accident duration had less effect. LA network, as usual, had a more reasonable increase since its streets were not as gridlocked as the other two networks. The significance of these results will allow us to test our solution explained in the next section.

V. Re-routing Mechanism Design

A. Key Principles

There were two options to reroute vehicles in SUMO. The first one consists in using statically defined routing mechanism provided by SUMO. This static method is deployed by adding a re-routing file to SUMOs configuration. All vehicles that need to be re-routed have to be listed in this file prior to simulation runtime. This method was tested using very small scale scenarios in which it worked well. However, since all the vehicles had to be defined prior to runtime it was not feasible to use this approach; the main reason being that in other scenarios there would be thousands of cars. This static method could be made dynamic by writing a script perhaps but the second re-routing option seemed more appropriate to simulate realistic road traffic and random accidents.

Combining SUMO with TraCI was the other option available; it involves using the server-client connection between the two and the TraCI API in python to alter the state of vehicles during simulation runtime. No vehicles had to be defined beforehand, only the road segment in which the accident will occur and the trigger roads surrounding it. When the initial testing of this re-routing with TraCI began, simple small scale tests were done, where the vehicles routes were manually defined so that the actions of the specific vehicles could be monitored. Once the strategy was working correctly a more dynamic approach was needed in order to perform large scale and multi scenario tests. The proposed re-routing strategy is explained by the state transition diagram shown in Figure 9.

The trigger mechanism was used to make our solution less complex as it consists in adding the road segments connected to the accidents lane to a list, and then comparing each vehicles current lane ID to each trigger to identify the vehicles that need to be re-routed. The alternative to this approach was to only reroute the vehicles that have the accident lane in their route definition. In order to deploy this alternative approach, the routes of all the vehicles have been written in a list (some routes contain up to a hundred edges), then the vehicles for which the route contains the accident lane have been identified by running a simple check. The complexity of this approach was quickly realised when the simulation steps exponentially increased in time as more vehicles deployed onto the network.

In Figure 10, the blue boxes highlight where the trigger lanes are while the yellow box shows the accident lane. When a car is found to be on a trigger lane its route is recalculated. During its recalculation the accident lane is excluded because it has a negligible speed limit (the speed limit is set to 0.001 meters per second to simulate a lane closure). The advantage of this approach is that it will not alter the route of vehicles that are on a trigger lane but don't intend to go onto the accident lane, because when the vehicles route is being recalculated, it doesn't have the accident lane in its route to start with, so the re-routing algorithm will just return its original route. Below is a pseudo-code example illustrating of how this mechanism works.

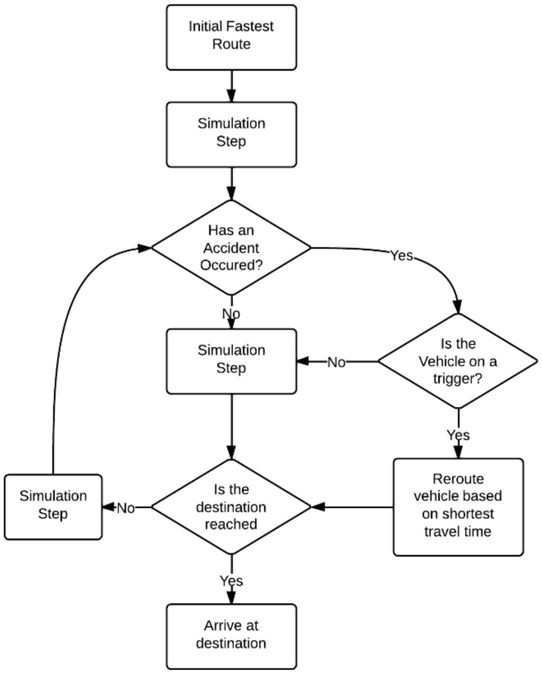

Figure 9: Flow Diagram illustrating the proposed re-routing strategy

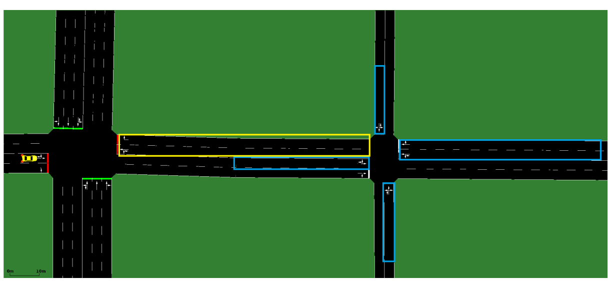

Figure 10: Trigger Lane Mechanism

```
def run():
    step = 0
    while step == 0 or
NumberOfVehicles() > 0:
        simulationStep()
        if step <= AccidentDuration:
            monitor()
            createAccident(laneID, speed)
        else if step > AccidentDuration:
            stopAccident(laneID, speed)

        step += 1

def monitor():
    trigger = [triggerLanes]
    listVeh = getIDList()
    for all vehicles in ListVeh:
```

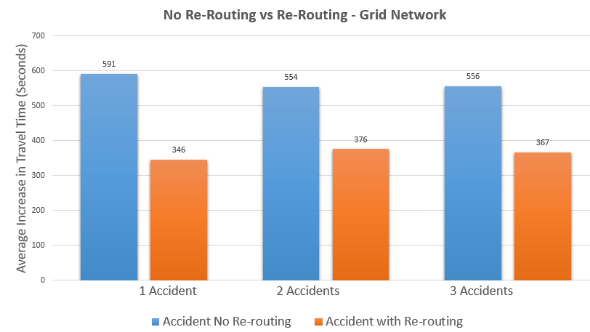

Figure 11: Our proposed re-routing mechanism efficiency: Grid network

```
        currentVeh = listVeh.pop()
        location = getRoadID(currentVeh)
        if trigger.contains(location):
            reroute(currentVeh)
```

B. Performance Evaluation Results

To evaluate the performance of our proposed re-routing mechanism, multiple tests and scenarios had to be run, similar to the accident testing above. For each scenario there were seven tests on a given traffic density. The seven tests involved were; no Accident (Just run the simulation), Accident but no re-routing (For one, two and three different accidents) and Accident with Re-Routing (also for one, two and three accidents). Despite being easier to manipulate the parameters of the tests using python and TraCI, the test runs took longer than just simulating an accident because for each time step all the vehicles locations had to be determined. Our implementation was of O(N) complexity (i.e. N = number of vehicles on the network), so when running more complex tests the simulation time increased. For the initial full scale test, the 10x10 Grid Network was used.

From the results shown in Figure 11, we can observe the improvement brought by our re-routing mechanism in case of one accident, which is mainly due to the small scale of the map. Moreover, our re-router still performs very well under two and three accidents. The overall average improvement of the re-router on the Grid Network is 35%, meaning that the average of 10 minutes delay has been reduced to 6.5 minutes only when our re-router mechanism is applied. A second test was carried out on LA road network, where the most positive results were collected. With its wide lanes and lack of one way streets it was the perfect scenario to show the strength of our proposed re-router.

The results depicted in Figure 12 indicate clearly that the re-router was significantly faster than the accident simulation; overall it shows an average of 60.5% improvement compared to the scenario without re-routing. The third test was carried out on New York network, from the offset it was assumed that New York would not provide the same level of improvement as the Grid or LA due to the frequency of one-way streets,

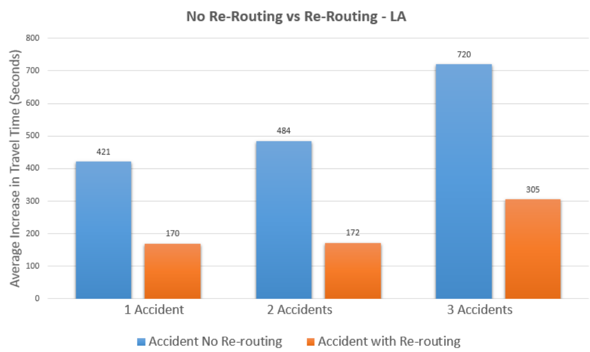

Figure 12: Our proposed re-routing mechanism efficiency: LA network

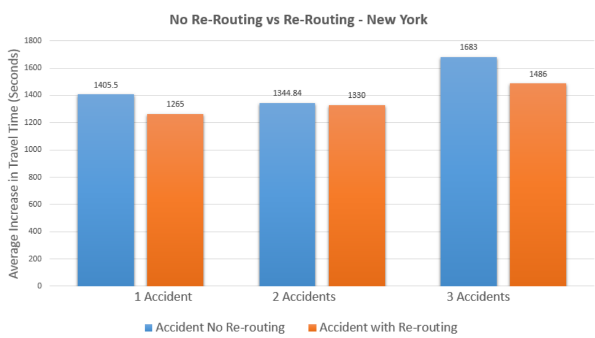

Figure 13: Our proposed re-routing mechanism efficiency: NewYork network

and the obtained results shown in Figure 13 have proven this assumption. Whilst the results were not as good as those of the previous two tests, they still showed an overall improvement of 7.6%. This could be improved on by altering the way in which vehicles were re-routed, by employing a different algorithm which would take into account the weight of certain nodes which are often beyond capacity in cities such as New York. If this weighting metric was added traffic could be diverted via the less frequently used nodes and thus alleviating congestion overall. It was found that in the NY simulation in particular, there were many collisions. As mentioned previously, collisions are bugs that are handled by teleporting, but there seemed to be an inordinate amount in NY, which caused further increases in travel time. This definitely affected the results of the re-router because vehicles would often get stuck behind a car that randomly collides with another, with no way of getting free from the jam it must wait until it is teleported. This increases the overall average increase in travel time because if a vehicle was re-routed and then it gets stuck in a jam (nothing to do with the manually inserted accidents), its travel time closely approaches or exceeds that of one which simply waits for the accident it was diverted from, to be cleared.

Our solution performed well on the above scenarios, but all the above tests used a fixed relatively short accident duration

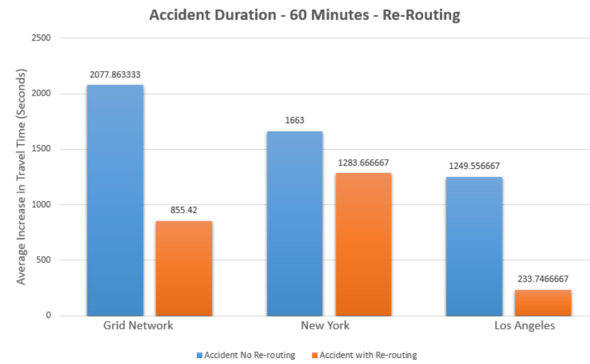

Figure 14: Our proposed re-routing mechanism efficiency: 60 minutes accident duration

of 20 minutes. To further test the efficiency of our solution, the accident duration was increased to 60 minutes. This meant that the re-router would have the chance to re-route much more traffic. The results shown in Figure 14 were obtained by running the same number of accidents per scenario as before, and averaging the increase in travel time from all accidents. The performance of the re-router on the 60 minute accident was better than the 20 minute accidents in all three scenarios, providing an overall average reduction of 53%. This equals to 33% improvement on the average reduction in travel time in the 20 minute tests, which was 35%. These tests proved the theory that the longer an accident is the more beneficial it is to try a different route. This parallels a real world scenario to a degree. If a driver was informed of traffic jam that would last half an hour, the driver would be much better off to follow another route that may add 20 minutes to the journey. The only thing that the driver does not know is whether or not an accident has occurred on the detour road they chose to take. This is what leads to further increases in travel time, which often happened in our simulations. To overcome these errors, we ran the accident for a longer duration, to illustrate the effectiveness of our solution. The main flaw with re-routing traffic is that the detour must take less time than the accident. If the accident is too short (e.g. less than ten minutes), more often than not the driver will end up worse off than if they decided to wait in the traffic jam.

From all the results discussed above, it is clear that the road network topology play a major role in the effectiveness of our re-routing solution. The more one way streets, single lane 2-way streets, traffic lights and junctions means the more complex a detour becomes. The histogram plotted in Figure 15 shows the average reduction in travel time under three different topologies. Even with a small sample of different topologies, major differences can be seen in the effectiveness of re-routing. This reflects the real world equivalent cities. As we would imagine it is easier to drive around LA than NY, and not just because of the number of cars on the road, but due to the shape and nature of the road networks.

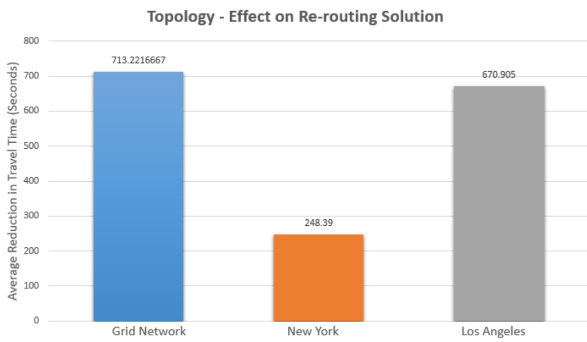

Figure 15: Impact of network topology on our solution efficiency

VI. Conclusion

The main goal of this paper was to extend SUMO by proposing a re-routing mechanism which dynamically updates the vehicles' route during simulation runtime to avoid the delay incurred by the occurrence of random accidents on the roads. This has been achieved through TraCI and its python API which was coupled with SUMO. After running numerous initial tests to highlight the effect of accidents on travel time, our proposed mechanism which updated the vehicles' routes during runtime was developed and further tests were run to evaluate its performance and prove its efficiency. These results revealed that the proposed re-router mechanism reduced the overall increase in travel time by an average of 35% (across all tests). Overall, the solution we employed performed very well on the given scenarios, further tests can be done to figure out how well it would work with real traffic demand such as TAPAS Cologne dataset[23]. As a future work, we intend to make the re-routing more dynamic by allowing accidents to be created on the fly and only decide on their duration. Moreover, we aim to consider more metrics in the design of the re-routing scheme such as toll costs, fuel consumption or easiness of driving.

VII. Acknowledgement

The authors would like to thank Mr. Shen Wang for his help and advices during the preparation of this work.
This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).

References

[1] R. Arnott & K. Small (1994), *The Economics of Traffic Congestion*, American Scientist, Volume 82(5), 446-455.
[2] url{http://www.transport2020.org/}
[3] P. Goodwin (2004), *The Economic Costs of Road Traffic Congestion*, Rail Freight Group, London UK, May 2004.
[4] S. Joerer, C. Sommer, and F. Dressler. (2012) *Toward Reproducibility and Comparability of IVC Simulation Studies: A Literature Survey*, VANET '12. ACM, NEW YORK, USA, 27-32.
[5] http://www.tomtom.com
[6] http://www.garmin.com
[7] http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/
[8] http://www.paramics-online.com
[9] http://www.aimsun.com/wp/
[10] J. Harri, M. Fiore and F. Fiali and C. Bonnet, *Vehicular Mobility Simulation with VanetMobiSim*, Trans. Soc. Modeling and Simulation, 2009.
[11] D. Choffnes & F. Bustamante *STRAW - An Integrated Mobility and Traffic Model*, In Proc. of the 2nd ACM International Workshop on Vehicular Ad-hoc Networks (2005).
[12] http://wiki.openstreetmap.org/wiki/TIGER
[13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. *Recent Development and Applications of SUMO - Simulation of Urban MObility. International Journal On Advances in Systems and Measurements*, 5 (3&4):128-138, December 2012.
[14] A. Wegner, M. Piorkowski, M. Raya, H. Hellbruck, S. Fischer and J.P. Hubaux. *TraCI: An Interface for Coupling Road Traffic and Network Simulators*,Proc. 11th Comm. and Networking Simulation Symp. (CNS 08), Apr. 2008.
[15] J. Pan, M. Khan, I. Popa, K. Zeitouni and C. Borcea. *Proactive vehicle re-routing strategies for congestion avoidance.*, 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems, Zhejiang, China, 16-18 May 2012.
[16] V. Ngoc Nha, S. Djahel and J. Murphy, *A Comparative Study if Vehicles' Routing Algorithms for Route Planning in Smart Cites*, VTM 2012, Dublin, Ireland, 20 Nov 2012.
[17] Chabini, Ismail, and Shan Lan. "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks." Intelligent Transportation Systems, IEEE Transactions on 3.1 (2002): 60-74.
[18] S. Wang, S. Djahel J. McManis, C. McKenna and L. Murphy. *Comprehensive Performance Analysis and Comparison of Vehicles Routing Algorithms in Smart Cities*. IEEE GIIS 2013, Trento Italy 28-31 October 2013.
[19] Dijkstra, Edsger W. "A note on two problems in connexion with graphs." Numerische mathematik 1.1 (1959): 269-271.
[20] http://www.inrix.com/
[21] B. Coifman and R. Mallika. *Distributed surveillance on freeways emphasizing incident detection and verification*, Transportation Research. Part A, Policy and Practice. Volume 41(8), 750-762.
[22] http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Students_And_Support
[23] http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Data/Scenarios/TAPASCologne