Framework for better Routing Assistance for Road Users exposed to Flooding in a Connected Vehicle Environment

Gaby Joe Hannoun

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Kevin Heaslip Co-Chair
Pamela Murray-Tuite Co-Chair
Kathleen Hancock

October 4th, 2017
Falls Church, Virginia

Keywords:
flooding, routing assistance, connected vehicles, in-vehicle navigation systems
Framework for better Routing Assistance for Road Users exposed to Flooding in a Connected Vehicle Environment

Gaby Joe Hannoun

ABSTRACT

Flooding can severely disrupt transportation systems. When safety measures are limited to road closures, vehicles affected by the flooding have an origin, destination, or path segment that is closed or soon-to-be flooded during the trip’s duration. This thesis introduces a framework to provide routing assistance and trip cancellation recommendations to affected vehicles. The framework relies on the connected vehicle environment for real-time link performance measures and flood data and evaluates the trip of the vehicle to determine whether it is affected by the flood or not. If the vehicle is affected and can still leave its origin, the framework generates the corresponding routing assistance in the form of hyperpath(s) or set of alternative paths. On the other hand, a vehicle with a closed origin receives a warning to wait at origin, while a vehicle with an affected destination is assigned to a new safe one. This framework is tested on two transportation networks. The evaluation of the framework’s scalability to different network sizes and the sensitivity of the results to various flood characteristics, policy-related variables and other dependencies are performed using simulated vehicle data and hypothetical flood scenarios. The computation times depends on the network size and flood depth but have generally an average of 1.47 seconds for the largest tested network and deepest tested flood. The framework has the potential to alleviate the impacts and inconveniences associated with flooding.
Flooding is a natural hazard that occurs with heavy rainfalls and high tides. In extreme situations, a flood in an area results in the evacuation of its occupant. Yet, in many cases, a flood is less severe and may only result in roads closures without necessitating evacuation. During these situations, and as transportation engineers, our ultimate goal is to maintain efficient and safe traffic operations. This thesis introduces a framework that focuses on providing routing assistance to affected vehicles and sending warnings to unaffected ones. It relies on the future connected vehicle environment which enables the communication between a traffic management center and equipped vehicles. The traffic management center collects and processes the information about the link performance measures and the weather and flood forecasts and sends them to the connected vehicles. Each vehicle has an in-vehicle navigation system in which the proposed framework is embedded. The framework, depending on the vehicle’s origin, destination, path and departure time and based on the flood’s characteristics, determines whether the vehicle is affected or not. If the vehicle is unaffected, it will receive a warning with the areas to avoid in case of any deviation and it can resume its trip as intended. If affected, the vehicle will either receive a warning to stay at its origin or routing guidance in the form of hyperpath or a set of alternative paths. The proposed framework has been evaluated on two transportation networks modeled in VISSIM based on the city of Virginia Beach, VA. Using simulated vehicle data and generated flood scenarios, several tests were executed to evaluate the scalability of the framework to different transportation networks along with the sensitivity of the results to variation in flood characteristics, policy-dependent variables and other dependencies. Concentrated, more intense and deeper floods resulted in a higher impact on the system. Yet, the analysis of the output is highly dependent on the location of the origin and destination of the vehicles with respect to the flooded roads. Thus, a lot of the output explanation are specific. Computation time increased with the increase in network size and in the flood depth. Nevertheless, it is still small and reasonable and further increase in both parameters (network size and flood depth) can be tested in future along with multiple techniques that minimize the
computation time. This framework addresses the flooding hazard which road users are experiencing more and more nowadays. This hazard brings risks and inconveniences to our daily life. Thus, the development of this framework is of great interest to our society as it is a promising tool that has the potential to offer benefits, in terms of safety and mobility, to roads users exposed to a flood hazard. Its first implementation shows that it is a timely application with a potential to perform even better with future improvements.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee co-chairs, Kevin Heaslip and Pamela Murray-Tuite, who offered me aspiring guidance and shared their constructive remarks and valuable ideas during these two years.

My genuine appreciation also goes to my undergraduate advisor John El Khoury, for his continuous encouragement and motivation and for offering me knowledge that has been a great asset to my graduate studies.

Last but not least, I would like to thank my amazing parents, brother and Johnny Aoude for their love, emotional support and enthusiasm. Thank you for keeping me smiling. I definitely could not have done this without you.
Table of contents

CHAPTER I: INTRODUCTION .................................................................................................................. 1
  1.1. Background ........................................................................................................................................ 2
  1.2. Goals and Objectives .......................................................................................................................... 4
  1.3. Contribution of this thesis .................................................................................................................. 4
  1.4. Organization of the thesis .................................................................................................................. 5

CHAPTER II: LITERATURE REVIEW ........................................................................................................ 6
  2.1. Dynamic route guidance systems ...................................................................................................... 6
  2.2. From the classical Dijkstra’s shortest path algorithm (1959) to the time-dependent
       Hyperstar algorithm (2012) ................................................................................................................. 9
  2.3. Emergency evacuation frameworks ..................................................................................................14
  2.4. Summary ............................................................................................................................................16

CHAPTER III: FRAMEWORK .....................................................................................................................17
  3.1. Overview .............................................................................................................................................17
  3.2. Terminology .......................................................................................................................................18
  3.3. Input data for the framework .............................................................................................................21
      3.3.1 Undelayed travel time .....................................................................................................................21
      3.3.2. Maximum delay ............................................................................................................................22
      3.3.3. Flood data .....................................................................................................................................22
  3.4. Framework methodology ..................................................................................................................24
      3.4.1. Buffer generation: .......................................................................................................................27
      3.4.2. Vehicle analysis ............................................................................................................................29
      3.4.3 Routing .........................................................................................................................................33
      3.4.4. Process .........................................................................................................................................34

CHAPTER IV: EXPERIMENTAL ANALYSIS ...............................................................................................44
  Test 1: Scalability of the framework for different network sizes and structures ....................................47
  Test 2 – Part a: Sensitivity of the results to various flood locations .........................................................54
  Test 2 – Part b: Sensitivity of the results to various flood depths ............................................................61
  Test 2 – Part c: Sensitivity of the results to various flood intensities .......................................................65
  Summary: Test 2 ........................................................................................................................................70
  Test 3 – Part a: The sensitivity of the results to various frequencies of weather and flood forecast
             updates ..................................................................................................................................................71
  Test 3 – Part b: The sensitivity of the results to various empty link intervals ............................................75
  Test 3 – Part c: The sensitivity of the results to various additional clearance intervals ............................78
  Summary: Test 3 .........................................................................................................................................81

CHAPTER V: CONCLUSION ..........................................................................................................................82

REFERENCES ..............................................................................................................................................85

APPENDIX A ..............................................................................................................................................90
List of figures

Figure 1: Flood boundary and buffers ................................................................. 19
Figure 2: Link timeline ....................................................................................... 19
Figure 3: Framework flowchart (future connected vehicle environment) ............. 24
Figure 4: Framework flowchart (research) ............................................................ 26
Figure 5: Input/Output data of buffer generation step .......................................... 27
Figure 6: Buffer generation pseudocode ............................................................ 28
Figure 7: Input/Output data of vehicles analysis step ........................................... 30
Figure 8: Vehicles analysis pseudocode ............................................................. 32
Figure 9: Group 1 ............................................................................................... 35
Figure 10: Group 2 ............................................................................................ 37
Figure 11: Group 4 ............................................................................................. 38
Figure 12: Group 5 ............................................................................................ 39
Figure 13: Input/Output data of process step ...................................................... 41
Figure 14: Coding ............................................................................................... 43
Figure 15: Network 1 in VISSIM (Baltic Ave. and 21st street) ............................ 44
Figure 16: Network 2 in VISSIM (Shore Drive Ave.) .......................................... 45
Figure 17: Results - Percentage of routed vehicles within each computation time interval for different networks (Test 1) ................................................................. 53
Figure 18: Scenario 1- Flood (yellow links) concentrated in the middle (Network 1 in VISSIM) ............................................................................................................ 54
Figure 19: Scenario 2- Flood (yellow links) concentrated in the middle (Network 1 in VISSIM) ............................................................................................................ 55
Figure 20: Results - Percentage of routed vehicles within each computation time interval per scenario (Test 2 a) ................................................................. 60
Figure 21: Percentage of routed vehicles within each computation time interval for different flood depths .................................................................................... 64
Figure 22: Time of flood (t4) distribution in each scenario .................................. 66
Figure 23: Percentage of routed vehicles within each computation time interval for different flood intensities ................................................................. 69
Figure 24: Percentage of routed vehicles within each computation time interval for different forecast update frequencies ......................................................... 74
Figure 25: Percentage of routed vehicles within each computation time interval for different empty link intervals ............................................................. 77
Figure 26: Percentage of routed vehicles within each computation time interval for different additional clearance intervals ................................................. 80
List of tables

Table 1: Dynamic route guidance systems ................................................................. 8
Table 2: Vehicle Groups ......................................................................................... 30
Table 3: Tasks .......................................................................................................... 40
Table 4: Base-case scenario parameters .................................................................. 46
Table 5: Network descriptors (Notation and Equation) ......................................... 48
Table 6: Networks characteristics .......................................................................... 48
Table 7: Results - Number of vehicles per category (Test 1) ................................. 50
Table 8: Results - Number of vehicles per group (Test 1) ..................................... 50
Table 9: Results - Computation times (Test 1) ....................................................... 52
Table 10: Results - Average computation times per group (Test 1) ....................... 52
Table 11: Results - Number of vehicles per category (Test 2 a) .......................... 57
Table 12: Results - Number of vehicles per group (Test 2 a) .............................. 57
Table 13: Results – Computation times for different flood locations (Test 2 a) ....... 59
Table 14: Results - Average computation times per group for different flood locations (Test 2 a) ................................................................. 59
Table 15: Results - Number of vehicles per category for different flood depths (Test 2 b) ................................................................. 62
Table 16: Results - Number of vehicles per group for different flood depths (Test 2 b) ................................................................. 62
Table 17: Results – Computation times for different flood locations (Test 2 b) ....... 63
Table 18: Results - Average computation times per group for different flood locations (Test 2 b) ................................................................. 63
Table 19: Flood time (t4) per elevation interval (Scenario 1) ................................. 65
Table 20: Flood time (t4) per elevation interval (Scenario 2) ................................. 65
Table 21: Flood time (t4) per elevation interval (Scenario 3) ................................. 65
Table 22: Results - Number of vehicles per category for different flood intensities (Test 2 c) ................................................................. 68
Table 23: Results - Number of vehicles and percentage of affected vehicles per group for different flood intensities (Test 2 c) ................................................................. 68
Table 24: Results – Computation times for different flood locations (Test 2 c) ....... 69
Table 25: Time of flood (t4) corresponding to each elevation interval in each scenario .... 71
Table 26: Results - Number of vehicles per category for different update frequencies (Test 3 a) ................................................................. 73
Table 27: Results - Number of vehicles per group for different update frequencies (Test 3 a) ................................................................. 73
Table 28: Results – Computation times for different flood locations (Test 3 a) ....... 73
Table 29: Time (t3) corresponding to each elevation interval in each scenario ......... 75
Table 30: Results - Number of vehicles per category for different empty link interval (Test 3 b) ................................................................. 76
Table 31: Results - Number of vehicles per group for different empty link interval (Test 3 b) ................................................................. 76
Table 32: Results – Computation times for different empty link intervals (Test 3 b) ...... 77
Table 33: Results - Number of vehicles per category for different additional clearance intervals (Test 3 c) ................................................................. 79
Table 34: Results - Number of vehicles and percentage of affected vehicles per group for different additional clearance intervals (Test 3 c) ................................................................. 79
Table 35: Results – Computation times for different additional clearance intervals (Test 3 c) ................................................................. 80
CHAPTER I: INTRODUCTION

Transportation engineers seek to make the road system resilient to natural hazards. Each road network, depending on its geographic location, is subject to specific threats. For example, regions near the Ring of Fire face numerous earthquakes while areas in the northern hemisphere and at tropical latitudes, such as the state of Florida in the US, are more prone to hurricanes [1]. Depending on the type of natural hazard, emergency evacuation plans should be developed to anticipate and lessen the impacts that a natural hazard has on road users. In this study, our focus is the flooding natural hazard that mostly occurs in regions at low altitude and near water.

According to Prada [2], three types of floods exist; flood due to insufficient capacity of the drainage system, flood due to sea level rise and flood due to river flowing over limited capacity. Flooding imposes a significant threat to the transportation network as it can cause traffic disruption when the water level reaches a certain threshold as well as possible structural damages, injuries and vehicle crashes. The effects of flooding can be classified as direct or indirect effects. Examples of direct effects are the repair costs of the damages while the costs of traffic disruption and inaccessibility to the affected area are examples of indirect effects.

Nuisance flooding, by definition, is a “flooding that leads to public inconveniences such as road closures—are increasingly common as coastal sea levels rise” [3]. It is important to note that once the flooding disappears, traffic operations cannot resume immediately since flooding often leaves behind layers of coarse grained soil along with long term damages that make the roads temporarily closed. Pyatkova et al. [4] assert that road closures due to flooding lead to the rerouting and/or the cancellation of affected trips. They highlight the fact that few studies have previously attempted to assess the impacts of floods on road networks and have assumed a static aspect of the transportation system which is unrealistic and hence leading to unreliable results. Subsequently, they developed a methodology that assesses the impacts of floods on road networks using microscopic traffic simulation. By comparing a base scenario to a flood scenario and by merging a microscopic traffic model with a flood model, they introduced a technique that determines the flood impacts on the road network by monetarizing intangible effects such as lost time, fuel consumption and emissions.

Since flooding leads to closure of the flooded roads, this study focuses on providing route assistance to the vehicles that are affected. Depending on their origin, and intended destination
and path, some vehicles are affected while others can proceed with their trip as planned. The affected vehicles have an origin, destination and/or intermediate node(s) on soon-to-be flooded or closed links. In these cases, in-vehicle route guidance systems can assist the users in leaving the danger zone and selecting the best alternative path. In addition, in-vehicle route guidance systems could communicate warning messages to unaffected vehicles alerting them about the areas to avoid.

1.1. Background

In-vehicle navigation systems were first developed in the 1960s when wireless communication technologies became available. They have also gained more attention in the 1990s with the rise of global positioning systems (GPS) that can provide reliable time and location information [5]. With the continuous advancements in communication technologies, the deployment of in-vehicle navigation systems implementing real-time route guidance systems is increasing. Nowadays, road users highly rely on route guidance systems to travel in urban transportation networks. According to Dong [4], these navigation systems have a significant influence on travel as they have the potential of alleviating congestion and improving the overall traffic performance. Generally speaking, this tool assists a road user by computing the optimal path between its origin and destination.

Different types of route guidance schemes exist. The first generation of route guidance systems are based on shortest path algorithms that result in a fixed path between two nodes in the network. One example of these shortest path algorithms is the well-known Dijkstra’s algorithm that will be discussed further in the next chapter. These route guidance systems are called static because static costs are attributed to the links in the network. In other words, link travel times are assumed to be fixed over time; hence, this type of route guidance systems does not capture variations in traffic conditions. Failing to consider the actual delays experienced on the roads, the static approach cannot guarantee the identification of the true optimal path.

To adapt to the varying traffic conditions, more advanced route guidance systems have been introduced and consist of computing the optimal paths in a transportation network with time-dependent link costs. This dynamic type of route guidance systems has gained attention after the tremendous advancement in communication technologies which allowed the estimation and broadcast of the real-time traffic measurements to be used in the optimal path generation. In
addition, after the initiation of the trip with the computed optimal path, the latter can become suboptimal due to unexpected incidents incurring significant travel delays. These dynamic route guidance systems have also focused on computing, based on the real-time information, en-route path adjustments. On the other hand, and according to Dong [5], dynamic routing guidance systems exist in two forms: centralized and decentralized. Centralized systems are more common and are based on the communication between vehicles and a traffic management center. The latter gathers real-time information to provide accurate traffic estimates to be used in route generation. However, in decentralized systems, the whole process is operated within the vehicle and is based on estimated link performance measures [5]; hence, less reliable route decisions are likely.

With the rise of the wireless and communication technologies, and as a part of the Intelligent Transportation System, the connected vehicle environment caught the attention of many researchers. A wide variety of studies focusing on enhancing the traffic operations on different levels emerged and the capabilities of the connected vehicle environment have been integrated in newly and previously developed studies to provide, in the transportation network, improved safety, enhanced mobility, and environmental benefits, among other benefits [6-8]. The connected vehicle environment relies on dedicated short range communication which allows for a rapid exchange of real-time and accurate information among vehicles and between vehicles and equipped infrastructure.

In this study, a dynamic route guidance framework for the routing of vehicles that are affected by a flood event is introduced. The centralized framework relies on the connected vehicle environment and assumes that real-time information and accurate traffic measurements are available and can be used to determine the actual link travel times, flooding progress, and delays. These link performance measures are assumed to be broadcasted from the traffic management center to the vehicles in which in-vehicle guidance systems are integrated. To compute a set of optimal alternative paths for the vehicles that need to be rerouted, the time-dependent hyperstar routing algorithm [9], discussed further in the next chapter, is adopted. The vehicles on soon-to-be flooded links are first directed out of the danger zone and then provided with a set of alternative paths to resume their trip to their original destination. The vehicles that
were heading towards a destination on a soon-to-be flooded or closed link are assigned to a new safe destination. A new set of alternative optimal paths are generated for vehicles with one or more soon-to-be flooded and/or closed link(s) in the path. Since the hyperstar routing algorithm computes a set of optimal paths between an origin/destination pair, the selection of the actual path depends entirely on the user’s preference. On the other hand, unaffected vehicles, present in the study area, receive a warning that includes the locations of all the roads and intersections to be avoided. Due to the connected vehicles’ capabilities, the traffic measurements are assumed to be continuously collected, allowing the regular update of link performance measures.

1.2. Goals and Objectives

This study focuses on three main goals. First, it provides a safer environment for vehicles in flood affected areas (Goal 1). Second, it enhances the route selection of users that are rerouted due to the road closures (Goal 2) and third, it informs all users of hazards (Goal 3).

To accomplish these goals, the following objectives are defined:

- Prevent vehicles from entering soon-to-be flooded and closed links and intersections (Goal 1)
- Rerouting of affected vehicles and warning sent to unaffected vehicles (Goal 1 and 3)
- Provide the user, who has to be rerouted, a set of optimal paths between an origin and a destination (Goal 2)
- Flexible framework set-up that adapts to various flooding scenarios (Goal 1)

1.3. Contribution of this thesis

A flood does not always necessitate the evacuation of all the occupants in the affected area. While people can still occupy the building, only roads that are expected to be flooded need to be emptied and the entry to them should be prevented. Previous literature has only focused on the flood as an extreme hazard necessitating evacuation of the affected area and has developed emergency evacuation frameworks for the routing of vehicles. Yet, this framework focuses on the flood natural hazard that is limited to road closures, such as from heavy rain in low lying areas. Routing of the vehicles in the event of a flooding resulting in road closures has not been explicitly addressed in the previous literature. This framework consists of routing affected
vehicles and sending warnings to unaffected ones. The framework does not require pre-planning
to determine the optimal strategy for each possible flood scenario. In fact, it can easily adapt to
any flood sequence and timeline.
Nowadays, in the event of a flood leading to road closures, the early closure of major roads that
are expected to be flooded in the near future is a way to ensure safety of road users. However,
this will result in excessive delays along with a significant number of vehicles that are stuck at
origins while the links that are closed could have been traveled for an additional time.
Nevertheless, there is a possibility of not closing some roads until water is obviously too high.
Hence, the decision of travelling through the road or not is left to personal judgement that will
impose safety threats on the road users.
The framework tries to minimize the delay that is currently being incurred through the use of in-
vehicle communication systems to continuously inform the users about the alternative routes to
their destination while preventing them to enter the roads that cannot be travelled anymore. In
other words, it relies on the rapid broadcast of real-time information and on the communication
between vehicles and a traffic management center available in the connected vehicle
environment, to close and prevent the entry to the soon-to-be flooded links only for safety as
needed.

1.4. Organization of the thesis

The remainder of this thesis is organized as follows. Chapter 2 comprises a review of
previous studies about routing of vehicles during emergency events and about existing dynamic
route guidance systems and routing algorithms. In chapter 3, the proposed framework along with
its assumptions and limitations is presented. Chapter 4 presents the results obtained after testing
the framework on two transportation networks based on Virginia Beach. Finally, chapter 5
summarizes the findings and concludes with the recommendations for future work.
CHAPTER II: LITERATURE REVIEW

This literature review chapter is divided into three sections. The first part includes a review of dynamic route guidance systems. The second part outlines how the time-dependent hyperstar routing algorithm, on which the proposed routing framework is based, has emerged from the traditional Dijkstra’s shortest path algorithm. The third part presents frameworks that focused on the routing of vehicles affected by a natural hazard.

2.1. Dynamic route guidance systems

There are two distinct forms of route guidance systems: the static route guidance systems that assume fixed link costs and the dynamic route guidance systems that assign each link in the network a varying link cost to calculate a more reliable path that takes into consideration the time-dependent link conditions. In this section, studies that focus on dynamic guidance systems are discussed.

Dynamic guidance systems require the collection of real-time information as input to the computations. Based on these data, the actual link performance measures or link travel costs are determined. As shown in Table 1, different techniques of data collection can be applied and the most common one relies on the use of the Intelligent Transportation Systems (ITS) [10-18]. The ITS program emerged to enhance the mobility and safety and lessen the environmental impacts of transportation systems and its scope includes the connected vehicle environment. In this modern transportation system, vehicles are equipped with sensors, GPS and wireless communication that allow the broadcast of real-time data about traffic conditions as well as the vehicles’ location, paths and destinations and that provide, in return, real-time traffic information about the links in their paths. Other link travel time estimation techniques are based on simulation using traffic measurements from inductive loops and image detection [19, 20].

Once the link performance measures are estimated using real-time traffic information, various methods for the dynamic computation of the optimal path can be implemented (Table 1). Wu et al., Nadi et al. and Fu [17, 18, 21] have a particular form of routing algorithm that does not compute the complete path from origin to destination all at once. In fact, the best path obtained based on real-time information before the initiation of the trip can turn into a suboptimal one during the trip due to changing link conditions. Thus, these studies find
progressively the best path to be adopted. Fu [21] is based on a dynamic program that identifies at each node the most optimal immediate link until the destination is reached. Travel times are not assumed to be fixed and can be estimated before entering the corresponding link. Similarly, in Nadi [18], a dynamic routing algorithm has been introduced and is based on a link-based strategy that selects the next best link found at each intersection. In Wu [17], the average travel time benefits of using real-time information to continuously adjust the path after the trip initiation by selecting the most optimal subsequent node during the current time interval are shown. Another dynamic en-route real-time (DEDR) route guidance scheme has been developed by Lin et al. [10] and consists of determining an optimal path at the origin and continuously adjusting it during the trip journey. The DEDR route guidance system determines for each link in the path a trust probability that takes into consideration the real-time link travel time and traffic density. This trust probability is compared to a specific threshold below which the link is not considered optimal and hence the generation of optimal alternative routes is triggered.

In [19], simulations were performed using real-time traffic measurements to compute the actual link travel times and traffic volumes. These link estimates were then used in a dynamic route guidance system with a multi-criteria objective function solved using a fuzzy decision model to replicate the various individual preferences involved during the path selection process. Similarly, and to replicate human decision making, a route choice model enhanced with real-time information that assists drivers in selecting the best path has been developed by Li et al. [22].
<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Path finding methodology</th>
<th>Real-time measurements for link travel time estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fu, Liping [21]</td>
<td>2001</td>
<td>Dynamic program</td>
<td>Assume that real time information is available random realization of travel time before entering the link</td>
</tr>
<tr>
<td>Wahle, Joachim, et al. [19]</td>
<td>2001</td>
<td>Multi criteria objective functions solved using fuzzy decision model</td>
<td>Simulation with real time traffic measurements from inductive loops</td>
</tr>
<tr>
<td>Lin, Jie, et al. [10]</td>
<td>2015</td>
<td>Link trust probability measure</td>
<td>ITS (GPS, sensor, wireless communication devices)</td>
</tr>
<tr>
<td>Chahbi, Ismehene, Dorra Ben Amara, and Abdelfettah Belghith [12]</td>
<td>2013</td>
<td>MVDR beamforming technique to maximize coverage while computing optimal path</td>
<td>ITS (roadside sensors and wireless communication devices, in-vehicle computer systems, V2R)</td>
</tr>
<tr>
<td>Li, Caixia, Sreenatha Gopalarao Anavatti, and Tapabrata Ray [22]</td>
<td>2014</td>
<td>Analytical hierarchy process (AHP) using a fuzzy inference technique</td>
<td>Prediction of future link travel time using a hybrid traffic prediction model based on historical data</td>
</tr>
<tr>
<td>Dong, Wei, Hai L. Vu, and Quoc Bao Vo. [14]</td>
<td>2011</td>
<td>Mathematical program using link travel time correlation</td>
<td>ITS</td>
</tr>
<tr>
<td>Tian, Daxin, et al. [15]</td>
<td>2013</td>
<td>Traditional Dijkstra algorithm</td>
<td>Real time state estimation by link division real time information from connected vehicles</td>
</tr>
<tr>
<td>Ding, J-W., et al. [16]</td>
<td>2010</td>
<td>RR packet dropping technique to determine potential alternative routes</td>
<td>ITS (Vehicle-to-Vehicle)</td>
</tr>
<tr>
<td>Wu, Chengjin, Xuedan Zhang, and Yuhan Dong. [17]</td>
<td>2013</td>
<td>Fastest path finding algorithm two time-adaptive decision rules based on greedy strategy</td>
<td>ITS (Connected Vehicles)</td>
</tr>
<tr>
<td>Nadi, Saeed, and Mahmoud Reza Delavar [18]</td>
<td>2010</td>
<td>Link-based strategy</td>
<td>ITS (sensors)</td>
</tr>
</tbody>
</table>
Unexpected incidents result in the sudden increase of travel times on specific links making the continuous adjustment of the path after the initiation of the trip highly advantageous. The benefits of real-time traffic information obtained through variable message signs, travel time displays, and dynamic route guidance to truck routing have been studied in [23]. Simulations have shown that dynamic route guidance is the most beneficial real-time traffic data communication form. Link travel times are regularly updated with the varying traffic conditions and the most optimal route from the current truck’s origin to its destination is adjusted using a shortest path algorithm. Similarly, and with the aim of enhancing traffic performance after an incident, a dynamic traffic assignment has been developed by Sawaya et al. [24]. The proposed methodology generates alternative routes around freeway incidents but after the implementation of a system optimum traffic assignment algorithm and control strategies algorithm. The system optimum traffic assignment algorithm minimizes the total network delay and computes the optimal flows in the routes around the incident. The control strategies algorithm identifies the alternative paths which are formed by the links on which flow changed. At each diversion, the recommended path is the one with the maximum flow. The set of all recommended paths is then evaluated using a simulator based on the cell transmission model and results indicated lower total travel time when compared to the do-nothing scenario.

2.2. From the classical Dijkstra’s shortest path algorithm (1959) to the time-dependent Hyperstar algorithm (2012)

Dijkstra’s algorithm [25] is a classical shortest path algorithm on which a wide variety of extensions were built. This algorithm computes a shortest path tree between an origin node and all destination nodes in a graph. The graph is directed and composed of a set of nodes and a set of non-negative weighted edges. Two lists are created: an open list that includes the nodes that have not been reached by the tree, and a closed list that includes all nodes that have already been expanded. At initialization, all nodes have a label set to infinity except for the origin node which is in the open list and whose label is set to zero. The node label refers to the tentative distance from the origin node to that node along the tree. From the open list and at each step, the node with the highest priority (i.e. smallest label) is selected, expanded and transferred to the closed list. The links that leave this selected node are added to the tree and the nodes that are reached are added to the open list and their corresponding labels are adjusted to the current tentative
distance from the origin along the tree. The algorithm terminates when the open list is empty. However, if only the shortest path between an origin and one destination node is needed, the algorithm can terminate when the destination node is reached and transferred to the closed list.

A wide variety of techniques focusing on speeding-up Dijkstra’s algorithm have emerged. They mainly consist of preprocessing data in the graph and then use the information to reduce the computational time while maintaining the same output which is the shortest path tree. Wagner and Willhalm [26] present a review of the Dijkstra’s speed-up techniques. They assert that these techniques consist of limiting the search space of the algorithm and they described two classical speed up techniques: the bidirectional search and the goal-directed search or Astar. The bidirectional search [27] is based on alternating between two unidirectional searches: one forward search from the origin node in the graph and one reverse search from the destination node in the reverse graph. When one node is transferred to the closed lists of both searches, the algorithm can be terminated and the shortest path between the origin and destination nodes is computed as the sum of the distance between the origin and the common node obtained from the normal search and the distance between the common node and the destination node obtained from the reverse search. On the other hand, the goal-directed search of Astar, which derived from the heuristic approach developed by Hart et al. [28], consists of adjusting the way the nodes are labeled thus affecting the order in which the nodes in the open list are expanded. In the unmodified Dijkstra’s algorithm, the node label reflects the distance from the origin to the node along the tree. However, the Astar speed-up technique adds to each node a potential or heuristic that depends on the desired destination node. For example, a node potential can be an underestimate of the remaining distance to the destination node [29]. The adequacy and accuracy of node potentials efficiently guide the search towards the destination node. The unmodified Dijkstra’s algorithm returns the shortest path from one node to all other nodes in a graph, but, as previously mentioned, it can also compute the shortest path to a specific node by terminating the algorithm when the latter is reached. The bidirectional search and the goal-directed search are techniques that focus on speeding-up the search for the shortest path between an origin and one destination node by efficiently reducing the search space thus guiding the algorithm’s search towards the destination node.
With the rise of the Intelligent Transportation Systems (ITS), route navigation systems are rapidly spreading. Shortest path methods originating from graph theory are being adapted and incorporated into the route guidance systems. Their objective is to alleviate traffic congestion and provide road users with optimal paths between an origin and a destination along the transportation network. Nevertheless, drivers who are provided with a single path from an origin to a destination can often end up on a suboptimal route. This is mainly due to the unreliability of the link travel times. In other words, link travel times are uncertain since traffic congestion and unexpected route events such as car accidents can significantly affect them. In addition, if all drivers are provided with the same shortest path, it becomes overloaded and hence not optimal with time [29]. Thus, the generation of multiple shortest paths from an origin to a destination is a technique that can improve the travel time reliability by spreading traffic on the different alternative paths while also taking into account driver’s preferences [30]. It can also be more efficient than the generation of a single shortest path in cases where unexpected link or node failures occur and rerouting is needed. Chen et al. [31] assert that multiple shortest paths algorithms in graph theory are mainly classified into two categories: the k-shortest path algorithms [31-42] and the totally disjoint path algorithms [43-45]. The latter consists of finding a number K of paths with minimum lengths joining the origin and the destination node but without sharing any intermediate node or link [45]. This class of algorithms is mainly utilized in communication networks and focus on always guaranteeing a path from the origin to the destination since a link failure on one link or node only affects one path and it is nearly impossible to have link failures on all paths simultaneously [43]. However, it is important to note that the totally disjoint path algorithms can often yield paths with unacceptable lengths and fail to include the primary shortest path.

On the other hand, k-shortest path algorithms can be classified on different levels. First, they can be categorized according to the output: there are algorithms generating simple paths (i.e. a node cannot be visited more than once) while others focus on finding looped paths (i.e. repeated nodes). K-shortest path algorithms can also be classified based on the type of technique used.

Traditional k-shortest path algorithms [32-34] are based on the enumeration of all possible paths up to a certain path length threshold followed by a ranking process. The main disadvantages of these algorithms relying on this technique are the significant computation time and the
considerable memory space needed that increase as the network size and the K value increase [31]. Van der Zijpp et al. [46] agrees that the enumeration of all paths, the deletion of infeasible paths and the ranking of the acceptable ones is an unrealistic technique. Thus, they enhanced this technique by the introduction of constraints that dictates whether a path is feasible or not. Thus, their improved technique, called the constrained K-shortest paths (CKSP), only leads to the enumeration of the feasible k-shortest paths (i.e. paths that satisfy a set of constraints).

Another group of k-shortest path algorithms are the ones that compute the k-shortest paths by adjusting the weights of the links in the k\textsuperscript{th} iteration’s shortest path [31, 35-37]. For example, in Yen [35], the k-shortest path is found by first setting the length of the links in the shortest path to infinity and then running the shortest path algorithm from the upstream node of each of these links to the destination node. Hence, deviations from the primarily shortest path can be obtained, facilitating rerouting. Other studies [31, 36, 37] have also focused on modifying the links’ weights on the shortest path but have adopted the link penalty method where the links’ weights are incrementally adjusted to find the k\textsuperscript{th} shortest path at each step.

The generation of a set of paths between an origin and a destination is an approach that gained attention since it has the potential to improve travel time reliability by accounting for the uncertainty of link travel times while also taking into consideration driver’s preference. The previously discussed path set generation algorithms compute a set of acceptable paths between an origin and a destination. Since the link travel times are considered static, the alternative paths can be ranked according to the fixed costs.

However, in the context of vehicle routing, the link travel times are not static and the uncertainty is reflected by experienced delay due to congestion during regular times of the day (peak and off-peak times) and/or irregular incidents. Therefore, Bell [29] developed a routing algorithm called Hyperstar that generates a reliable path set. It is an reinterpretation of the Spiess and Florian algorithm [47] which is designed for transit assignment. The latter determines, for each link, a service frequency that is equal to the inverse of the waiting time on the link. Next, it finds a hyperpath between an origin and a destination (set of possible optimal paths) by minimizing the expected travel time to the destination. Bell adapted the Spiess and Florian algorithm to vehicle
routing by considering the service frequency of a link equal to the inverse of the maximum delay that can be experienced on a link. For instance, a link with a high maximum delay has a low service frequency and is hence less reliable. Since the hyperstar algorithm is designed for vehicle routing systems, it incorporated the Astar algorithm to speed up the generation. Ma et al. [48] introduced the Dijkstra-Hyperstar algorithm which incorporates two techniques to speed-up the Hyperstar algorithm while maintaining the same generated hyperpath. It used Dijkstra’s algorithm to determine the node potentials and adopted a node directed search to limit the number of links being evaluated.

Another method that considers the dynamic aspect of networks was introduced by Chabini et al. [49] and consists of assigning to each link a time-dependent cost. So, as an extension to the work of Chabini et al. [49] in which a single path is computed between a pair of nodes based on link travel times that vary with time, Bell et al. [9] revisited the hyperstar algorithm discussed above and added to it a dynamic and more realistic aspect. An undelayed travel time along with a maximum delay are assigned to each link and are both dependent on the time of arrival at the link. The algorithm reversed the original hyperstar algorithm in which the travel time of a link depends on the time of departure from a link. In fact, the time-dependent hyperstar algorithm builds the hyperpath from the origin to the destination by minimizing the expected arrival at the destination and all intermediate nodes.

In this study, we implement the time-dependent version of the hyperstar algorithm [9] to provide a set of alternative paths to the vehicles affected by the flooding; thus, enhancing their route selection while maintaining their preferences. During a flood event, the state of the system (i.e. road closures) as well as the link performance measures vary with time. For instance, a vehicle can travel a link that will become closed and unavailable for other vehicles in the future. So the time-dependent hyperstar algorithm is used because it is a path finding method that can adapt to the dynamic aspects of the flood event and the transportation system itself by assigning for each link a time-dependent travel time and maximum delay.
2.3. Emergency evacuation frameworks

Early detection of flooding can help in reducing its impacts on road users by broadcasting warnings and implementing effective evacuation strategies. Detection becomes more challenging in the case of flash flood which refers to a rapid and no-notice rise and fall of water levels [50]. To forecast spatial and temporal flood progression, a wide variety of flood prediction models have been developed [51]. Inputs to these hydrological models are accurate precipitation measurements. According to Liechti [51], obtaining quantitative precipitation estimates (QPEs) from the weather radar is a technique that can provide real-time rainfall data like the one obtained from the rain gauges. Other common techniques for rainfall data collection are the numerical weather prediction systems. Developing a reliable flooding forecast model is very challenging as uncertainty in the rainfall data cannot be completely eliminated. Output from flood prediction models include spatial and temporal distributions of the flooding at the street-level in the transportation network.

In extreme conditions, a flood event requires the evacuation of all the occupants of the affected area. As a result, previously established emergency evacuation plans need to be implemented. To identify effective plans for each event, a wide variety of evacuation traffic simulation models, not specific to any natural hazards, have been introduced and these models differ on several levels. According to Stepanov et al. [52], macro level evacuation models are models that consider traffic as uninterrupted flow while the meso or micro level models perceive traffic as platoons of vehicles or individual vehicles respectively, making the integration of the sociological and psychological factors more possible [53]. Regarding the route assignment technique, some models adopt a static procedure that only provides road users with one evacuation route while other models consider a more dynamic and realistic aspect where users have to make choices at intersections.

Two main approaches for evacuation models exist. The first consists of defining the optimal paths in the network and then evaluating their corresponding performance measures to select the best one for each origin-destination pair. The second approach is based on determining optimal routing policies using analytical techniques and then simulating its implementation to determine its feasibility and efficiency during emergency evacuations. According to Stepanov et al. [52], the former approach has been extensively studied and the second one needs more attention. Thus, they developed a multi-objective integer programming model that finds the
optimal evacuation routing policy for regional emergency evacuation. Their approach is based on combining analytical and simulation (MGCCSimul) techniques. After determining, for each origin-destination pair, the set of possible paths using the kth shortest path algorithm, the model generates a route assignment plan that minimizes the total clearance time, travelled distance and congestion while assuming a M/G/c/c queueing model that takes into consideration the decline in links’ service rate with the increase in traffic.

Similarly, other studies [54, 55] showed that simulation techniques can be widely adopted to evaluate different evacuation plans and to determine the best one for each scenario. For example, increasing public transportation ridership along with the use of contraflow lanes resulted in the most efficient and smooth operations for the studied transportation network.

On the other hand, an evacuation model specific for flood disaster has been developed by Chanta et al. [56]. This study highlighted the fact that in extreme situations only special vehicles with the ability to travel on roads with high water levels will have to transport people from the flooded areas to the safe areas. Thus, they developed a routing approach for these special vehicles to satisfy the time-dependent demands with the least possible number of vehicles. It is based on a mixed integer programming model that minimizes the total travelled distance. It assumes that vehicles have a maximum capacity and only operate during specific time intervals.

During emergency evacuations, it is important to consider the emergency response vehicles’ operations since these vehicles have to reach their destination as quickly as possible since they are most of the time involved in life-threatening situations that cannot be delayed. To provide them with increased mobility and accessibility when travelling to or from the evacuation zones, several studies have developed emergency evacuation models focusing on identifying the best routes for emergency response vehicles [54, 57-59]. In Yu et al. [54], the use of simulation techniques to determine whether an evacuation strategy ensures efficient operations of the emergency response vehicles is presented. According to Oran et al. [58], routing models for emergency response vehicles consist of determining first the optimal facility locations at which the vehicles should be deployed (depots) and next the best routes to be travelled. One way to identify the facility locations is using the maximum coverage location problem while the vehicle routing problem with time windows can be utilized to determine the best routes between depots and destinations. In Oran et al. [58], priority considerations have been integrated in the facility location problem and the vehicle routing problem. In other words, when an emergency occurs, its
severity is not homogenous among the affected areas. Thus, Oran et al. [58] adjusted the formulations to compute the optimal deployment locations and routes to serve the areas with higher priority first. Similarly, Goel et al. and Jotshi et al. [6, 59] develop a methodology for optimal dispatching and routing of emergency response vehicles during a post-disaster evacuation. It uses data fusion to replicate real-world road conditions and hence to compute more reliable evacuation plans. In Chen et al. [57], a real time-dependent shortest path algorithm is introduced to identify the best paths between the depots and the demand points while minimizing the total travel time and the time of arrival at destination. To replicate reality, link travel times are regularly updated and the routes are subsequently adjusted to adapt to the varying road conditions. With this dynamic aspect, the algorithm provides real-time emergency routing for no-notice emergency evacuations.

2.4. Summary

Dynamic route guidance systems have more attention especially after the rise of the ITS that made the fast broadcast of accurate real-time traffic information possible. The hyperstar is a time-dependent routing algorithm developed by Bell [9]. It was theoretically introduced to route one vehicle from an origin to a destination and was tested only for demonstration purposes on a sample network that does not replicate real transportation networks.

In the proposed framework, the time-dependent hyperstar algorithm is used as the dynamic route guidance system to route vehicles during a flood event. The severity of the flood event that is considered in this study does not require the evacuation of buildings in the affected area, but only the prevention of entry of vehicles to the soon-to-be flooded and closed roads. The hyperstar algorithm is hence integrated in a framework that routes the affected vehicles that are still capable of departing from their origins. The consideration of intersections in the transportation network along with techniques for link data retrieval are examined to make the framework applicable on real road networks. Most importantly, the proposed framework leverages the use of the connected vehicles technologies, which provide the real-time information about the link conditions and affected vehicles to compute the optimal paths and to generate the required warning messages for vehicles in the affected area.
CHAPTER III: FRAMEWORK

In this chapter, the proposed framework is discussed. Its objectives, assumptions and methodology are also presented.

3.1. Overview

This framework relies on the connected vehicle environment in which communications among equipped vehicles and between equipped vehicles and infrastructure are enabled. Real-time traffic information from these vehicles is collected and processed to provide reliable performance measures to be used for traffic optimization purposes.

This framework leverages the benefits of the connected vehicle environment by first collecting the information about the vehicles (origin, destination and path) detected in the system and about the links and intersections in the network (travel time, delay, open/closed status). Since a set of roads in the transportation network will be flooded and subsequently closed, vehicles which were located on one of these links and vehicles which intended to travel through one will require assistance in determining their new route to reach their destination safely. This study focuses on routing the affected vehicles just prior to and during a flood. The connected vehicles also allow the broadcast of warning messages to the unaffected vehicles.

The framework acts as a centralized dynamic route guidance system during flood emergency situations. A Traffic Management Center (TMC) is assumed to gather all traffic information from the connected vehicles. The TMC estimates and regularly updates the current link performance measures and the flood impacts at the street-level. In-vehicle systems generate the routing guidance after the receipt of all required information from the TMC.

The framework does not require the generation of optimal emergency evacuation strategies prior the emergency event. The objective here is to develop a framework that adapts to flood scenarios with different flood timelines and locations.

The vehicle routing algorithm used in this framework is the time-dependent hyperstar algorithm introduced by Bell (2012) [9], which computes a set of optimal paths, called a hyperpath,
between an origin and a destination. For faster hyperpath computation, the algorithm requires the generation of node potentials to direct the search towards the destination. In this study, a node potential refers to the remaining undelayed travel time from the node to the destination and is calculated using the Partitioning shortest path algorithm [60]. The hyperstar algorithm is a theoretical routing algorithm that has been previously applied on a sample network for demonstration purposes [9]. In this study, the hyperstar is tested on more realistic transportation networks that account for the wait time experienced at signalized intersections.

The following assumptions were made when developing the framework:

- If a vehicle can travel, it will travel.
- If a link is flooded, its start and end nodes can still be travelled.
- A link is either open or closed (no partial closure).
- Only affected vehicles are rerouted.
- Unaffected vehicles receive a warning message that includes the areas to avoid.
- Vehicles with origins on closed links receive a warning message to wait at the origin.
- A traffic management center is available to process the real-time information obtained from the connected vehicles and to estimate the current link travel times and delays as well as road closures due to the flood.
- In-vehicle systems generate the route guidance (set of alternative paths) based on the link performance measures and flood information broadcasted from the traffic management center.
- Vehicles with destinations on soon-to-be flooded or closed links are assigned to new safe destinations where they can wait.
- Vehicles that cannot travel to any safe destination can wait at their origins.

3.2. Terminology

This subpart defines the terms used throughout the remainder of the thesis.

The first set of terms categorizes nodes near the flooding hazard.

- Flood boundary (Figure 1): The flood boundary includes the downstream nodes of the soon-to-be flooded links and closed links. These nodes can be connected to a safe node in Buffer 1 by a safe link.
• Buffer 1 (Figure 1): Buffer 1 includes the safe (unflooded) nodes that are connected to one node in the flood boundary by a safe link.

• Buffer 2 (Figure 1): Buffer 2 includes the safe (unflooded) nodes that are connected to one node in Buffer 1 by a safe link and are farther away from the hazard area than nodes in Buffer 1.

![Figure 1: Flood boundary and buffers](image)

• Link timeline (Figure 2):
The next set of terms pertains to the timeline of events for a link that experiences flooding. These are illustrated in Figure 2. Not all links in the network experience flooding; some links are considered safe for the entire time. Only links that flood require the below designations.

![Figure 2: Link timeline](image)

• Flood interval: This is the time interval during which the link cannot be travelled and is closed due to water on the road.

• Empty link interval: This interval acts as a safety margin during which the link is closed. No vehicles should be on the link during this time.
• Minimum clearance (and additional clearance) interval: this is the time interval during which the link is considered as soon-to-be flooded and only vehicles that are in the soon-to-be flooded zone can travel on these links.

• Safe interval: this is a time interval during which the link is not threatened by flood water, is considered safe, and can be travelled by any vehicle.

• Closed link: A link cannot be traversed when it is closed. A link that experiences flooding is closed at time $t_3$ (Figure 2) and thereafter.

• Soon-to-be flooded link: A link that will be flooded in the future is considered soon-to-be flooded before it actually floods. In Figure 2, the soon-to-be-flooded designation is assigned to the relevant links between time $t_1$ ($\geq t_1$) and $t_3$ ($<t_3$).

• Link time of closing: The time of closing is the time after which the entry to the link is prevented and it varies from an individual vehicle’s perspective depending on the vehicle’s initial position. For instance, vehicles with origins on soon-to-be flooded links are allowed to travel along soon-to-be (but not yet) flooded links to reach a safe stop, while vehicles originally on safe links are prohibited from entering soon-to-be flooded links. Thus, for vehicles outside the soon-to-be flooded and/or closed area, the time of closing of a soon-to-be flooded link is at time $t_1$ in Figure 2, while time of closing of the same link is at time $t_3$ for vehicles that are located in the soon-to-be flooded area.

Also referring to Figure 2, an affected vehicle is a vehicle with an initial or regular path that is directly impacted by the flood or safety measures. The affected vehicle meets at least one of the following criteria:

• Has an origin on a soon-to-be flooded link ($t_1 \leq$ departure time $< t_3$) or on a closed link (departure time $\geq t_3$)

• Is heading to a destination on soon-to-be flooded or closed link (expected time at destination $\geq t_1$)

• Intends to travel (based on the initial or regular path) through one or more intermediate soon-to-be flooded or closed link (expected time of arrival at the link $\geq t_1$).

The hyperstar algorithm as originally developed, uses the terms undelayed travel time and maximum delay [9, 29]. In the context of this study, these terms are considered as follows:
• Undelayed travel time: This is the link travel time that varies with the regular time of traffic conditions. Peak travel times are higher than off-peak but still considered undelayed in the absence of adverse weather or incidents.

• Maximum delay: This is the maximum delay incurred on a link due to the flood and is set to a very large number (infinite) to represent the closed condition (i.e. it is infinite after the link time of closing).

3.3. Input data for the framework

The input data required for the framework includes:

• Undelayed link travel times. This information is required for all links in the network and is assumed to be available from a traffic management center either from current technology (e.g., sensors/detectors) or from connected vehicles.

• Maximum delay. This information is required for all links in the network. This link performance measure can be obtained from sensors, detectors or using the connected vehicles as long as the link can still be travelled. The maximum delay is infinite when entry to the link is prohibited.

• Flood input data. For research purposes, a list of links that will flood with their expected time of flooding is needed. In future deployment, similar information could be obtained from weather and flood forecast systems.

In this research, outputs from VISSIM simulations along with generated flooding scenarios constitute the input data required for the numerical implementation of the framework in Chapter 4.

3.3.1 Undelayed travel time

To estimate each link’s undelayed travel time, the transportation network was modeled in PTV VISSIM software. Simulations were performed with different link flow conditions to generate travel time records for the vehicles in the network. These records were used to develop a travel time function representing the undelayed travel time for each link. As shown in equation 1, the undelayed link travel time is the dependent variable and the independent variable is the flow $Q$. In addition, the intercept is the free flow travel time.

$$Undelayed \ Link \ Travel \ Time_n = Free \ Flow \ Travel \ Time_n + \beta_1(Q)^2$$ (1)
In urban transportation networks, signalized and unsignalized intersections exist. A vehicle approaching an intersection can experience delays due to the red signal, stop sign or yield sign at the intersection. In other words, the vehicle has to wait on the link upstream of the intersection before proceeding through the intersection when granted the green light. The waiting time incurred varies with the different downstream movements that the vehicle intends to follow (i.e., right turn, left turn or straight movement). The total waiting time experienced on the upstream link is not accounted for in the undelayed travel time of that link. However, the waiting time that corresponds to each movement is assigned to the connector of that movement. Connectors are the virtual links in an intersection that connect all the different approaches. For instance, a four-way signalized intersection has twelve virtual connectors.

3.3.2. Maximum delay
For research purposes, the maximum delay experienced on a link while travel is still permitted is estimated based on information from the Federal Highway Administration. According to the Federal Highway Administration [61], a heavy rain event leads to an estimated free-flow speed (FFS) reduction between 6% and 17%. This is reflected by an increase in the free-flow time (FFT) between 6.5% and 20.5% for the unflooded links in this study. Thus, if the vehicle is expected to enter the link before the link closing time, the maximum delay experienced on the link is a duration between 6.5% and 20.5% of the FFT. However, if the time of entry is after the link closing time, the maximum delay is infinite and as a result the entry to the link is indirectly prevented.

3.3.3. Flood data
Information about the flood in terms of location and timeline in the transportation network are crucial inputs to this framework. We assume that these data can be obtained from future weather and flood forecast systems. For this study, flood information is generated to create and test various flood scenarios. Information includes the links and intersections that are expected to flood along with the time of flooding (time “t4” in Figure 2). Note that when an intersection is flooded, all of its connectors are considered flooded as well.
To ensure a margin of safety, a link that is expected to be flooded and unavailable for travel after a specific time has to be closed earlier (time “t3” in Figure 2). The empty link interval represents the time interval during which vehicles are not allowed to travel in any direction along the link. This variable would be determined by policy makers or a transportation agency and can vary with the link characteristics. However, in this study, it is considered fixed for all links.

The minimum clearance and additional clearance interval (from time “t1” to time “t3”) is a time interval during which only vehicles that are originally in the flooded area are allowed to travel (enter and/or exit links). Once these vehicles reach a safe zone (i.e., exit the soon to be flooded zone), they are denied re-entry. The link minimum clearance interval (from time “t2” to time “t3”) depends on the link characteristics, the traffic and weather conditions. The additional clearance interval (from time “t1” to time “t2”) is an interval whose duration is determined by policy makers or a transportation agency based on how safe and sufficient the minimum clearance interval is.

To sum up, at time t1, entry to the soon-to-be flooded zone is prohibited and only vehicles originally in this area are allowed to travel within it in order to exit and reach a safe stop. Thus, at time t1, barriers that block the entry points to the flooded zones should be placed.

At time t3, no vehicle is allowed to travel on the link. Therefore, barriers that close the entry and exit points of all flooded links should be positioned.

Note that a link is considered soon-to-be flooded during the minimum clearance and additional clearance interval. Vehicles initially located outside the flooded zone are not allowed to travel along soon-to-be flooded and closed links, and the time of closing of these links is considered at time t1. Vehicles that are originally inside the flooded zone can travel along soon-to-be flooded links, and the links’ time of closing, for these vehicles, is at time t3.
3.4. Framework methodology

This framework has been developed to be implemented in a connected vehicle environment. As shown in the flowchart in Figure 3, once a vehicle in the transportation network is detected, its trip is evaluated based on the actual state of the transportation network (i.e. current location of the flooded links) to determine whether the vehicle is affected by the flood or not. The remaining trip of an en-route vehicle is evaluated and its origin is considered its position at the time the framework is activated. However, the complete trips of new vehicles in the system are evaluated. If the vehicle is affected but is still able to leave its origin, it receives routing assistance to reach its final destination safely and as soon as possible. If the vehicle is affected but cannot reach a safe destination, it receives a warning stating that it should wait at the origin. In case the vehicle is unaffected, a warning notifies the vehicle about the areas to avoid. The actual link data are assumed to be estimated, stored and regularly updated in the traffic management center’s database through the broadcast of real-time information from the connected vehicles and updates from weather and hydrologic models. Vehicles in the system are evaluated independently. To determine if the vehicle is affected or not, along with the corresponding route guidance or warning to be displayed, the in-vehicle route guidance system retrieves the current link data from the traffic management center.

![Figure 3: Framework flowchart (future connected vehicle environment)](image-url)
In this research, and to demonstrate the implementation of this framework with simulated vehicle data, another flowchart has been adopted and is illustrated in Figure 4. Instead of assuming that the link data (i.e. travel time and information about flooding) are provided from the TMC, link performance measures were obtained from microscopic simulations of a transportation network modeled in VISSIM and information about the flood was generated for hypothetical scenarios. All vehicles recorded during a one-hour simulation in VISSIM are evaluated all at once based on their complete trips. Then, the vehicles are considered successively and depending whether they are affected or not, a vehicle either receives a warning or route guidance.

In the remainder of this chapter, the four major parts of the framework are discussed. First is the “buffer generation” part which explains how the safe node buffers around the flooded links are defined. These node buffers act as first safe stops or final destinations to some groups of vehicles. Second is the “vehicle analysis” in which the vehicles detected in the system are evaluated and grouped. The third part introduces the “routing” procedure which consists of executing the hyperstar routing algorithm and the partitioning shortest path algorithm whenever route guidance between two nodes in the network is needed. In the fourth part called “process”, the tasks specific to each group of vehicles are discussed.
Figure 4: Framework flowchart (research)
3.4.1. Buffer generation:

This step consists of generating two node buffers around each flood boundary. These buffers are defined above in section 3.2.

As shown in Figure 5, this step requires as inputs the network connectivity (link ID, upstream node, and downstream node) and the lists of links that are soon-to-be flooded or closed. In future connected vehicle environments, the node buffers would be updated every time the state of the system changes. In other words, the input list of the links is regularly updated so that the current node buffers are defined.

![Figure 5: Input/Output data of buffer generation step](image)

This step is coded using the Java computer programming language and the corresponding pseudocode is shown in Figure 6. Note that in future connected vehicle environment, the node buffers can be identified by computing the buffers at specific distances from the flooded locations.
Buffer generation main class {
    Read the Network input file (link id, from node, to node)
    Read the input_FloodedLinks file (list of soon-to-be flooded and closed links)

    Find the nodes in Flood Boundary(FB) {
        For (all links in input_FloodedLinks) {
            Add the origin and destination to the FB array;
        }
        For (all nodes in FB array) {
            If (node is not an origin or destination to a safe link)
                Remove node from FB array;
        }
        Remove duplicates in FB array;
        Sort FB array in ascending order;
    }
    Output: list of nodes in FB array

    Find the nodes in Buffer 1{
        For (all safe links) {
            If (origin is in FB && destination not in FB)
                Add destination in buffer 1 array;
            If (destination is in FB && origin not in FB)
                Add origin in buffer 1 array;
            If (origin is in FB && destination not in FB) or (destination is in FB && origin not in FB)
                Add link to Link_buffer array;
        }
        Remove node in FB from buffer 1 array;
        Remove duplicates in buffer 1 array;
        Remove duplicates in Link_buffer array;
        Sort nodes in buffer 1 array in ascending order of ID;
    }
    Output: list of nodes in Buffer 1;

    Find the nodes in Buffer 2 {
        For (all safe link && links not in Link_buffer array) {
            If (origin is in buffer 1 && destination not in buffer 1)
                Add destination in buffer 2 array;
            If (destination is in buffer 1 && origin not in buffer 1)
                Add origin in buffer 2 array;
        }
        Remove nodes in FB and in buffer 1 from the buffer 2 array;
        Remove duplicates in buffer 2 array;
        Sort buffer 2 array in ascending order;
    }
    Output: list of nodes in Buffer 2;
}

Figure 6: Buffer generation pseudocode
3.4.2. Vehicle analysis

Once the links that are soon-to-be flooded or closed are defined, all vehicles in the area of interest are evaluated to identify the affected and unaffected vehicles. Note that the buffer generation and the vehicle analysis steps are independent and can be executed in parallel.

As defined in section 3.2, an affected vehicle is a vehicle with an origin, destination or intermediate link that is soon-to-be flooded or closed within the duration of the trip. A vehicle that does not satisfy any of these conditions is not considered affected by the flood. In other words, its origin, destination and path are not expected to be soon-to-be flooded or closed during the trip.

Regarding the affected vehicles, if they can still leave their origins (origin not on a closed link at the departure time) and if at least one path can be found between their position and a safe destination, they receive route assistance that comprises a set of alternative paths from which the user selects his/her preferred one. On the other hand, affected vehicles that cannot travel only receive a warning instructing them to wait at the origin. All unaffected vehicles only receive a warning that discloses the areas to avoid in case of deviation from the planned path.

The vehicle data obtained from the one-hour microscopic traffic simulation with VISSIM, includes, for each vehicle, the list of links in the path and the time of entry to each link. In this step, and as shown in Figure 7, we extract from the raw data obtained from the VISSIM simulation, the origin, destination and trip start time for each vehicle. In addition, we classify the vehicles into six groups based on the flooded links and corresponding closing time, as shown in Table 2.
Table 2: Vehicle Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Priority</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>Vehicle with an <strong>origin</strong> on a link that is soon-to-be flooded at departure time and a <strong>destination</strong> on a link that is soon-to-be flooded or closed at the expected arrival time</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Vehicle with an <strong>origin</strong> on a link that is soon-to-be flooded at departure time</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Vehicle with an <strong>origin</strong> on a link that is closed at departure time or vehicle that cannot reach any safe destination</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Vehicle with a <strong>destination</strong> on a link that is soon-to-be flooded or closed at the expected arrival time</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Vehicle with an intermediate link that is soon-to-be flooded or closed at the expected time of entry</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Unaffected vehicles</td>
</tr>
</tbody>
</table>

In future connected vehicle environment, all vehicles in the system at the time the framework is initiated, would be analyzed and grouped simultaneously. Then, once a new vehicle is detected in the area, it will be analyzed and grouped independently.

![Figure 7: Input/Output data of vehicles analysis step](image)

A java code has been developed to complete this step and the pseudocode is shown in Figure 8. As shown in Table 2, each group is given a specific priority that goes from 1 to 6, with 1 being
the highest. A vehicle can belong to more than one group but the one with the highest priority dominates. For example, if a vehicle belongs to group 5 and group 3, the latter dominates because the vehicle cannot leave its origin and should only receive a warning to wait at the origin. Similarly, if a vehicle satisfies the conditions of group 2 and group 4, it is assigned to group 1.

This hierarchy is ensured through the order of the if-statements in the grouping method shown in the vehicle analysis pseudocode (Figure 8). Each if-statement evaluates the conditions of a given group. The order adopted goes from the lowest to the highest group priority. Hence, all vehicles are first assigned to group 6 which is the group with the lowest priority. Then, if the vehicle satisfies the conditions of the next if-statement’s group (i.e. group with higher priority), the group will be over written to obtain at the end the final group of each vehicle.
Vehicle analysis main class {

    Input: Raw vehicle file (number of links in the path for each vehicle; Vehicle raw data from vissim)
    Input: FloodedLinks file
    Input: FloodedNodesLinks_time (list of time at which each flooded link is closed or "t3")
    Input: Network connectivity (link id, from node, to node)
    Input: CL (duration of the minimum clearance and additional clearance interval in min)

For each vehicle v {
    v.group(input_FloodedLinks, FloodedNodesLinks_time, CL);
    v.origin();
    v.destination();
    v.start_time();
}

method group (input: FloodedLinks, FloodedNodesLinks_time, CL) {

    group=6; //Vehicle unaffected

    if (intermediate link in the path is soon-to-be flooded or closed && expected arrival at this link >= t3 - CL)
        group=5; //Vehicle with intermediate link soon-to-be flooded or closed

    if (first link is soon-to-be flooded && (t3-CL <= trip start time < t3))
        group=2; //Vehicle with origin on soon-to-be flooded origin

    if (last link is soon-to-be flooded or closed && expected time of arrival at destination >= t3- CL)
        group=4; //Vehicle with destination on soon-to-be or closed link

    if (group 2 && group 4)
        group=1; //Vehicle with origin on soon-to-be flooded link and destination on soon-to-be or closed link

    if (first link in path is closed && start_time >= t3)
        group=3; //Vehicle with origin on closed link

    return group;
}

method origin() {
    returns start node of first link in path;
}

method destination() {
    returns end node of last link in path;
}

method start_time() {
    returns time of entry at first link in path;
}

Figure 8: Vehicles analysis pseudocode
The routing algorithm implemented in this framework is the time-dependent hyperstar routing algorithm developed by Bell [9]. As discussed in chapter 2, this algorithm computes a hyperpath which is a set of alternative optimal paths between an origin/destination pair while speeding up the search with the use of node potentials as in the Astar algorithm [28]. The hyperstar algorithm requires, for each link, an undelayed travel time (Section 3.3.1.) and a maximum delay (Section 3.3.2) and the latter accounts for the road closures. In future connected vehicle environment, the link performance measures would be updated regularly through the connected vehicles and TMCs. Once the routing algorithm is triggered, the most recent link data is used. In this research, we assigned, for each link, a time threshold after which the link is considered closed and the delay is infinite to reflect the flood.

Depending on the vehicle’s group, the vehicle either receives a warning or route guidance. When the latter is needed, the hyperstar algorithm is initiated and requires, as input, a set of node potentials that is specific to the destination. In our study, a potential at a node refers to the remaining undelayed travel time from the node to the destination.

To obtain all node potentials to a specific node in the network, the partitioning shortest path algorithm [60] is used. This algorithm computes the shortest path from one node to all other nodes in the network. Thus, computing all node potentials to a given destination consists of first, reversing the network (i.e., reversing the direction of each link while keeping the same link costs), and then implementing the algorithm from the destination to all nodes.

For translation to the field, we assume that the node potentials to each node in the network are already computed and stored in the traffic management center’s database. As a result, whenever the hyperstar from an origin to a destination is to be implemented, the node potentials that correspond to the destination are retrieved and used to ensure faster computations of the hyperpath. In this research, once the hyperpath between an origin and a destination has to be identified, the partitioning algorithm is initiated and the node potentials to the given destinations are computed before calculating hyperpaths.
3.4.4. Process

In this final step, vehicles are evaluated successively. Once selected, and depending on the vehicle’s group, the corresponding tasks are executed as follows:

- **Group 1 (Figure 9):** These vehicles have an origin on a link that is soon-to-be flooded at departure time and a destination on a link that is soon-to-be flooded or closed at the vehicle’s expected arrival time. The objective is to assist them first in reaching a safe stop as soon as possible, and then in resuming their trip towards a final destination. These vehicles cannot reach their original destination because their expected time of arrival falls after the time “t1” (refer to Figure 2) at which the link is soon-to-be flooded or closed. Thus, a new safe destination will be proposed to the driver. It is a node in buffer 2 that is the closest to the original destination in terms of travel time. The driver will have to either accept the new destination or reject it and select another safe destination. In this study, we assume that all users accept the proposed new destination in buffer 2. Once the new destination is defined, its closest node from buffer 1 is then identified and acts as the first safe stop. For instance, the vehicle is routed from its origin to the closest node in buffer 1 first, to ensure it exits the flooded zone as soon as possible while also minimizing its remaining distance to the final destination. During this part of the trip, the vehicle is allowed to travel on soon-to-be flooded links but entry to closed links is prohibited. Once the vehicle reaches the intermediate safe stop at buffer 1, the vehicle is provided with a new set of alternative paths to reach its final safe destination while preventing re-entry to the soon-to-be flooded and closed links. When no hyperpath is found between the vehicle’s position and the assigned first stop, the next closest node in buffer 1 is selected. If no path exists between the vehicle’s origin and any node in buffer 1, the vehicle has to wait at the origin and is, hence, moved to group 3. When no hyperpath is found between the node in buffer 1 and the final destination, we assume that the user can wait at buffer 1.
Figure 9: Group 1
• Group 2 (Figure 10): These vehicles have an origin on a link that is soon-to-be flooded at departure time. The goal is to assist them in reaching a safe stop first before resuming the trip toward the final destination. The closest node from buffer 1 to the destination is identified and acts as the first safe stop. A set of alternative paths is computed between the origin and the node in buffer 1 (this hyperpath may include soon-to-be flooded links but not closed links), then a new set of paths will be identified from the node in buffer 1 to the final destination (this hyperpath cannot include soon-to-be flooded or closed links).

When no hyperpath is found between the vehicle’s initial position and the closest node in buffer 1, the next closest node to the destination is evaluated. If no hyperpath is found between the position and any node in buffer 1, the vehicle has to wait at the origin and is listed in group 3. When no hyperpath is found between the node in buffer 1 and the destination, the user waits at buffer 1.

• Group 3: These vehicles have an origin on a link that is closed at departure time. Thus, they receive a warning message instructing them to wait at the origin. This group includes as well the vehicles from groups 1, 2, 4 and 5 that cannot reach any safe destination.

• Group 4 (Figure 11): These vehicles have a destination on a link that is soon-to-be flooded or closed at the vehicle’s expected time of arrival. A new destination at the closest node in buffer 2 to the original destination is assigned. The user subsequently receives a set of alternative paths between its origin and the new safe destination (this hyperpath cannot include soon-to-be flooded or closed links).

When no hyperpath exists between the origin and the selected node in buffer 2, the next closest node to the destination is evaluated. If no hyperpath can be identified between the origin and any of the nodes in buffer 2, the vehicle has to wait at the origin and is listed in group 3.
Figure 10: Group 2
Figure 11: Group 4

Group 4

Run Partitioning from destination

Find the closest node in buffer 2 to destination

New safe destination = closest node in buffer 2

Run Partitioning from selected node in buffer 2

Run dynamic hyperstar between origin and node in buffer 2

hyperpath found?

Yes

Select a path

No

Next closest node in buffer 2 exists?

Yes

Vehicle moved to group 3

No
• Group 5 (Figure 12): These vehicles have one or more intermediate links that are soon-to-be flooded or closed at the expected time of entry. As a result, the driver has to modify his/her planned path. The driver is provided with a set of alternative paths between the original origin and destination (this hyperpath cannot include soon-to-be flooded or closed links). If no hyperpath exists between the vehicle’s position and the destination, the vehicle waits at the origin and is listed in group 3.
• Group 6: These vehicles are not affected by the flood. Their origin, destination, and path are not expected to flood or to be considered as soon-to-be flooded within the duration of the trip. As a result, they just receive a message that alerts them about the locations to be avoided in case of any deviation from the planned trip.

Table 3 summarizes the tasks that correspond to each group of vehicles. Affected vehicles that can still travel receive routing assistance (vehicles in groups 1, 2, 4 and 5), whereas affected vehicles that cannot reach any safe destination wait at the origin (group 3) and unaffected vehicles receive warning messages disclosing the areas to avoid (group 6).

Table 3: Tasks

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| 1     | Vehicle with an **origin** on a link that is soon-to-be flooded at departure time and a **destination** on a link that is soon-to-be flooded or closed at the expected arrival time | a. Set of paths from origin to node in buffer 1 *(hyperpath may include soon-to-be flooded links but not closed links)*  
b. Set of paths from node in buffer 1 to node in buffer 2 *(hyperpath cannot include soon-to-be flooded or closed links)*  
or wait at buffer 1 |
| 2     | Vehicle with an **origin** on a link that is soon-to-be flooded at departure time | a. Set of paths from origin to node in buffer 1 *(hyperpath may include soon-to-be flooded links but not closed links)*  
b. Set of paths from node in buffer 1 to destination *(hyperpath cannot include soon-to-be flooded or closed links)*  
or wait at buffer 1 |
| 3     | Vehicle with an **origin** on a link that is closed at departure time  
or vehicle that cannot reach any safe destination | Send warning: “WAIT at origin” |
| 4     | Vehicle with a **destination** on a link that is soon-to-be flooded or closed at the expected arrival time | Set of paths from origin to node in buffer 2 *(hyperpath cannot include soon-to-be flooded or closed links)* |
| 5     | Vehicle with a **intermediate** link that is soon-to-be flooded or closed at the expected time of entry | Set of paths from origin to destination *(hyperpath cannot include soon-to-be flooded or closed links)* |
| 6     | Unaffected vehicles | Send warning: areas to avoid |
As shown in Figure 13, the output files of the buffer generation and vehicle analysis steps are the inputs of the process step. The output files of the process step are:

- A file that includes, for each vehicle that has been routed, the corresponding hyperpath along with the total travel time and the computation times in java
- A list of the vehicles in each group

For translation to the field, since the route guidance system is assumed to be an in-vehicle system, the generation of the hyperpaths for each vehicle is executed independently, after the retrieval of the corresponding current link data and potentials from the traffic management center’s database. In this research, the vehicles in the system have been evaluated sequentially. Since the hyperstar generates multiple paths between an origin and a destination, we randomly select one path to simulate the human behavior. However, the path selection would be left to the user in future connected vehicle environment.

The pseudocode of this step is shown in Appendix A.
In Figure 14, the four parts along with the corresponding inputs and outputs are illustrated. In the next chapter, the framework will be implemented on two transportation networks in Virginia Beach, VA. The networks were modeled in PTV VISSIM software by Tony Fuentes and Fatema Siddiquee. The objective is to test the scalability of this framework for different network sizes and structures, to test the sensitivity of the output to different flood characteristics and various policies and dependencies.
Figure 14: Coding
CHAPTER IV: EXPERIMENTAL ANALYSIS

In this chapter, the proposed framework is implemented on two transportation networks to test its applicability and sensitivity of its output to flood characteristics, network size and structure, and policy variables.

The two transportation networks are based on flood-prone locations of Virginia Beach, VA. The first network is built around Baltic Ave. and 21st street, a location that floods every time there is a big storm (3 times in summer 2014) [62]. The second area includes part of Shore Drive Ave. and is considered a critical area during flood events since it encloses a fire department with high risk of being unusable due to high water levels. The transportation networks (Figure 15 and 16) were modeled with the PTV VISSIM microscopic traffic simulation program by Tony Fuentes and Fatema Siddique for the MATS-UTC project.

Figure 15: Network 1 in VISSIM (Baltic Ave. and 21st street)
The objectives of this experimental analysis are to evaluate:

1. The scalability of the framework for different network sizes and structures (Test 1)
2. The sensitivity of the results to various flood characteristics
   a. Location (dispersed/concentrated) (Test 2-part a)
   b. Depth (Test 2-part b)
   c. Intensity or speed of the flood (Test 2-part c)
3. The sensitivity of the results to policy related variables and other dependencies
   a. Frequency of weather and flood forecast updates (near-continuous, incremental) (Test 3-part a)
   b. Empty link interval (Test 3-part b)
   c. Additional clearance interval (Test 3-part c)

Note that Test 2 and 3 and executed only on Network 1, while both networks are used in Test 1. All the tests were executed on a MacBook Pro with a 2.7 GHz Intel Core i5 processor and a 16 GB 1867 MHz DDR3 memory.
The base-case scenario parameters are as shown in Table 4.

*Table 4: Base-case scenario parameters*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Network 1 (Baltic Ave. and 21st street)</td>
</tr>
<tr>
<td>Flood location and depth</td>
<td>Links at 2 m elevation and below</td>
</tr>
<tr>
<td>Frequency of weather and flood forecast updates</td>
<td>Every 10 minutes</td>
</tr>
<tr>
<td>Timing of flood</td>
<td>Elevation &lt;= 0.5 m: ( t_4 = 20 ) min</td>
</tr>
<tr>
<td></td>
<td>0.5 m &lt; Elevation &lt;= 1 m: ( t_4 = 30 ) min</td>
</tr>
<tr>
<td></td>
<td>1 m &lt; Elevation &lt;= 1.5 m: ( t_4 = 40 ) min</td>
</tr>
<tr>
<td></td>
<td>1.5 m &lt; Elevation &lt;= 2 m: ( t_4 = 50 ) min</td>
</tr>
<tr>
<td>Empty link interval</td>
<td>5 min</td>
</tr>
<tr>
<td>Minimum clearance interval</td>
<td>5 min</td>
</tr>
<tr>
<td></td>
<td><em>(In Baltic network: max link travel time = 2.22 min)</em></td>
</tr>
<tr>
<td>Additional clearance interval</td>
<td>5 min</td>
</tr>
</tbody>
</table>
Test 1: Scalability of the framework for different network sizes and structures

In this test, the framework is executed on Network 1 and Network 2 to test the scalability of the framework. The objective of this test is to evaluate the framework on two different network sizes and structures to evaluate the efficiency of the framework in terms of computational times. This comparison indicates how sensitive the computation times are to the network size and structure and whether the framework can be executed on larger networks while maintaining reasonable computation times.

As shown in Figures 15 and 16, the two networks have different sizes and structures. Network 1 is larger than Network 2 with approximately 9 times more links and 8 times more nodes. In addition, Network 2 has a more linear shape while Network 1 is more round. When visually inspecting the two test networks, we can also notice that Network 1 has a higher percentage of local streets when compared to Network 2. The latter has a major arterial lying along the entire network from east to west. However, comparing the two networks visually is not sufficient. For instance, there are several measures that can reveal important network characteristics to develop a proper network analysis and comparison. The network connectivity is a key aspect when evaluating a transportation network. A connected network is a network in which a path exists between any pair of nodes and a completely connected (or complete) network is a network in which every pair of nodes is connected by a link. There are several descriptors that reflect the degree of connectivity as well as the degree of complexity of a network. Complex networks are the large networks that exist in real life such as in transportation systems [63]. First, the cyclomatic number is a measure that indicates the number of independent cycles that exist in a network. It is computed using equation (2). Network 1 has a higher cyclomatic number than Network 2, yet this is mainly due to the fact that Network 1 is significantly larger than Network 2. The beta index is computed using equation (3) by dividing the total number of links by the total number of nodes. A tree network which is a connected network with no cycles has a beta index less than 1 while a network with only one cycle has a beta index equal to 1. Network 1 and network 2 both have a beta index higher than 1 and this indicates that the networks are complex networks and hence representing real-life systems. Regarding the connectivity of the networks, the alpha index is a measure that reflects the number of cycles with respect to the maximum possible number of cycles in the network. The higher the alpha index, the more connected the
network is. Similarly, the gamma index is a descriptor that evaluates the total number of links with respect to the total number of possible links. A higher gamma index implies a more complete network. As shown in Table 6, Network 2 has a higher gamma index as well as a higher alpha index, thus, Network 2 is more connected and more complete than Network 1.

Table 5: Network descriptors (Notation and Equation)

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Notation</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of links</td>
<td>$A$</td>
<td></td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>$N$</td>
<td></td>
</tr>
<tr>
<td>Number of isolated subgraphs</td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>Cyclomatic number</td>
<td>$\mu$</td>
<td>$\mu = A - N + p$</td>
</tr>
<tr>
<td>Beta index</td>
<td>$\beta$</td>
<td>$\beta = A/N$</td>
</tr>
<tr>
<td>Alpha index</td>
<td>$\alpha$</td>
<td>$\alpha = 2\mu/(N-1)(N-2)$</td>
</tr>
<tr>
<td>Gamma index</td>
<td>$\gamma$</td>
<td>$\gamma = 2A/(N(N-1))$</td>
</tr>
</tbody>
</table>

Table 6: Networks characteristics

<table>
<thead>
<tr>
<th>Network characteristics</th>
<th>Network 1</th>
<th>Network 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of links</td>
<td>6632</td>
<td>743</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>4442</td>
<td>572</td>
</tr>
<tr>
<td>Number of flooded links</td>
<td>178</td>
<td>20</td>
</tr>
<tr>
<td>Cyclomatic number</td>
<td>2191</td>
<td>172</td>
</tr>
<tr>
<td>Beta index</td>
<td>1.49</td>
<td>1.3</td>
</tr>
<tr>
<td>Alpha index</td>
<td>0.00022</td>
<td>0.00106</td>
</tr>
<tr>
<td>Gamma index</td>
<td>0.00067</td>
<td>0.0045</td>
</tr>
<tr>
<td>Number of vehicles</td>
<td>5002</td>
<td>1984</td>
</tr>
</tbody>
</table>

In both networks, the same percentage of flooded links with respect to the total number of links is tested. Since the base-case scenario implies that 2.7 % ($=178/6632$) of the links in Network 1 are flooded, the 20 ($=(743*178)/6632$) lowest-lying links are considered flooded in Network 2.
RESULTS

As shown in Table 7, the number of affected vehicles in Network 2 is slightly higher than the one in Network 1. Yet, the percentage of affected vehicles in Network 2 is 6.8% (=135/1984) and is significantly higher than the one for Network 1 which is equal to 2.2% (=111/5002). The same percentage of flooded links is tested; however, the impact of the flood is more extensive in Network 2. In fact, the location of the flooded links relative to the location of the origins and destinations of the vehicles in the system affects the number of affected vehicles and is the main reason behind the difference in the numbers of vehicles per category (shown in Table 7) even when the same percentage of links is flooded. In addition, the functional classification of the flooded links plays a major role in defining the impact of the flood on the vehicles. For instance, it is obvious that a flood on an arterial or a highway affects more vehicles than a flood on a residential and local street. This is due to the fact that a higher functional category of transportation links has higher capacity and can accommodate more vehicles.

In Table 8, the initial and final number of vehicles per group is shown for each network. The initial grouping is the grouping that results from the “vehicle analysis” step in the framework while the final grouping is the one obtained after executing the “process” step. Obtaining different initial and final grouping occurs when vehicles that have to be routed (vehicles in group 1, 2, 4 or 5) are moved to group 3 after the “process” step. These vehicles that are moved to group 3 are not located on a closed link at departure time but are vehicles that cannot reach a safe destination because of downstream floods which prevent the generation of any possible hyperpath between their origin and destination. Regarding Network 1, the initial and final groupings are the same. However, in Network 2, 2 vehicles and 3 vehicles are moved from group 4 and group 5, respectively, to group 3 while vehicles in group 1 and group 2 have all received routing assistance.

On the other hand, as shown in Table 8, the affected vehicles are distributed differently across the groups in the two networks. For instance, no vehicle belongs to group 1 in Network 1 while in Network 2, 3% (= 4/135) of the affected vehicles are added to group 1 which represents the vehicles with a soon-to-be flooded origin and a soon-to-be or closed destination. Furthermore, 9% (=10/111) of the affected vehicles belong to group 2 (i.e. vehicles with origins on soon-to-be flooded links) in Network 1 while 76% (=103/135) of the affected vehicles belong to group 2 in Network 2. This significant difference is due to the fact that in Network 2, the flood occurs on
links on which vehicles are generated in the system. In addition, 10% (=11/111) of the affected vehicles belong to group 3 and have to wait at the origin in Network 1; however, in Network 2, 0% of the affected vehicles have origins on closed links (initial grouping) but 3.7% (=5/135) were moved to group 3 because no hyperpath was found. In Network 2, no vehicle belongs to group 3 in the initial grouping because no vehicle is generated from the links with vehicles inputs after their corresponding time t3 while 103 vehicles (number of vehicles in group 2 in initial grouping) were generated between times t1 and t3 of these links that are considered soon-to-be flooded at the vehicles’ departure time. Regarding group 4 (i.e. vehicles with soon-to-be flooded destinations) and group 5 (i.e. vehicles with soon-to-be flooded or closed intermediate link(s)), higher percentages per group are recorded in Network 1 compared to Network 2. It is difficult to draw a general rule based on the distributions of affected vehicles among groups. The grouping of vehicles highly depends on the relative position of the flooded links to the position of the volume generation points in the modeled network (in VISSIM). In real life, a flood that leads to the closure of major production and attraction links as well as a flood on higher capacity links are expected to affect the system more than floods on links with low productions and attractions and floods on local roads.

*Table 7: Results - Number of vehicles per category (Test 1)*

<table>
<thead>
<tr>
<th>Category</th>
<th>Network</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unaffected vehicles</td>
<td></td>
<td>4891</td>
<td>1849</td>
</tr>
<tr>
<td>Number of affected vehicles</td>
<td></td>
<td>111</td>
<td>135</td>
</tr>
<tr>
<td>Number of routed vehicles</td>
<td></td>
<td>100</td>
<td>130</td>
</tr>
</tbody>
</table>

*Table 8: Results - Number of vehicles per group (Test 1)*

<table>
<thead>
<tr>
<th>Group</th>
<th>Initial grouping</th>
<th>Final grouping</th>
<th>Initial grouping</th>
<th>Final grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Group2</td>
<td>10</td>
<td>10</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>Group3</td>
<td>11</td>
<td>11</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Group4</td>
<td>63</td>
<td>63</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>Group5</td>
<td>27</td>
<td>27</td>
<td>13</td>
<td>10</td>
</tr>
</tbody>
</table>
According to Table 9, the average computation time as well as the minimum and maximum computation times recorded in Network 2 are lower than the ones in Network 1. Similarly, all average computation times per group are smaller for Network 2. These results are expected because Network 2 is smaller than Network 1 and it is obvious that the computations of the hyperpaths take less time because the hyperpaths in Network 2 are by default shorter than the ones in Network 1. The group with the highest average computation time is group 4 in Network 1. This group includes the vehicles with soon-to-be flooded or closed destinations at their expected arrival time. This group recorded the highest average computation time due to the time-consuming search for a new destination. For instance, the maximum computation time of 49.04 seconds belongs to a vehicle (ID: 4394) in group 4 and it is the search for a new destination that requires 48.95 seconds. In real life, and in order to minimize the significant computation times required for the search for a new destination for vehicles in group 1 or group 4 (i.e. vehicles with soon-to-be flooded or closed destinations), the search for a new destination can be stopped after evaluating a specific number of tentative destinations that turned out to be inaccessible. In future implementation, the framework can also be extended to allow the users in groups 1 and 4 to enter new destinations of their choice; thus, enhancing the user’s preferences and the computational efficiency by eliminating the search for a new destination. In Network 2, the group with the highest average computation time is group 2. The computation times for this group is the sum of the computation times needed to generate a hyperpath from the origin to a node in buffer 1 and the computation times needed to generate a hyperpath from buffer 1 to the destination. In Network 2, the average computation time is very small which means the search for new safe destination is less time-consuming in smaller networks.

As shown in Figure 17, in Network 1, the computation time interval that included the highest percentage of routed vehicles is the interval between 0.5 and 1 second. Three vehicles in group 4 recorded a computation time greater than 2 seconds; these vehicles are assigned to a new destination and it is the search for a new accessible destination in buffer 2 that is time-consuming. In Network 2, the computation times for all routed vehicles fall in the interval between 0 and 0.5 second. The reason why the computation times in Network 2 are significantly smaller than in Network 1 is because the former is considerably smaller. It is the time needed for the search for a new destination for vehicles in group 4 that is the most sensitive to the size of the network. This test indicates that the increase of the network size results in an increase of the
required computation times recorded by routed vehicles. Yet, the computation times recorded in Network 1, except for the vehicles that recorded a computation time greater than 2 seconds, are still reasonable and can be minimized by stopping the search after scanning a specific number of tentative destinations. Further increase in the network size can be tested in the future.

Table 9: Results - Computation times (Test 1)

<table>
<thead>
<tr>
<th>Computation times (seconds)</th>
<th>Network</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td></td>
<td>1.47</td>
<td>0.06</td>
</tr>
<tr>
<td>Min</td>
<td></td>
<td>0.39</td>
<td>0.01</td>
</tr>
<tr>
<td>Max</td>
<td></td>
<td>49.04</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 10: Results - Average computation times per group (Test 1)

<table>
<thead>
<tr>
<th>Average computation times (seconds)</th>
<th>Network</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td></td>
<td>1.20</td>
<td>0.07</td>
</tr>
<tr>
<td>Group 2</td>
<td></td>
<td>1.88</td>
<td>0.05</td>
</tr>
<tr>
<td>Group 4</td>
<td></td>
<td>0.61</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Figure 17: Results - Percentage of routed vehicles within each computation time interval for different networks (Test 1)
Test 2 – Part a: Sensitivity of the results to various flood locations

This test consists of executing the framework for Network 1 with different flood locations i.e. with different sets of flooded links. Weather and flood forecast systems define the set of links that will flood with their expected time of flood \( t_f \). It is important to analyze the transportation network prior to a flood event and to locate the areas with high productions, attractions and the roads travelled by large traffic volumes. This allows for a broad understanding of how significant a flood in a given location is. The objective of this test is to determine the effect of changing the location of the flood within the network.

The following scenarios are tested:

- **Scenario 1:** Flood concentrated in the middle of the network (Number of flooded links = 178)

![Figure 18: Scenario 1 - Flood (yellow links) concentrated in the middle (Network 1 in VISSIM)]

- **Scenario 2:** Flood concentrated on the side of the network (Number of flooded links = 178)
• Scenario 3: Dispersed flood (Number of flooded links = 178)

For comparison purposes, the flooded areas in scenarios 1 and 2 are defined in a way to enclose the same number of flooded links (178) as in scenario 3 and all links in all scenarios are assumed to flood at 20 min ($t_4 = 20$ min).

RESULTS

As shown in Table 11, scenario 2 (side of the network) resulted in the highest number of affected vehicles (2112) followed by scenario 1 (938) then scenario 3 (273). Hence, in Network 1, a flood that is concentrated in one location affects the network significantly more than a dispersed flood. Furthermore, in the case of the dispersed flood, 86% ($=234/273$) of the affected vehicles are routed while in the cases where floods are concentrated at the middle and on the side of the network, 60% ($=568/938$) and 69% ($=1458/2112$) received routing assistance respectively.
Consequently, in this network, a dispersed flood has less impact on the network as less vehicles are required to wait at the origin. Yet, the impact that a specific set of flooded links has on a network depends on the location of the origins and destinations and departure times of the vehicles in the system. Therefore, the results of this test are specific to this network and cannot be generalized.

As shown in Table 12, the initial and final grouping of the affected vehicles in scenario 3 (dispersed) are similar. However, after executing the “process” step in scenario 1 and 2 (concentrated flood), the distribution of affected vehicles among groups changed. All vehicles in groups 1 and 2 in scenarios 1 and 2 received routing assistance (same number of vehicles per group in initial and final grouping). Yet, in scenario 1, 2.6% (= (421-410)/421) of the vehicles in group 4 and 16.5% (= (91-76)/76) of the vehicles in group 5 are moved to group 3 because no hyperpaths to their destinations were found. Similarly, in scenario 2, 0.4% (= (741-738)/741) of the vehicles in group 4 and 23.9% (= (758-577)/758) of the vehicles in group 5 are moved to group 3 and forced to wait at the origin along with vehicles that have an origin on a closed link at the time of departure (vehicles in group 3 in the initial grouping). Consequently, with concentrated floods, more vehicles have to wait at the origin. These vehicles are not only the ones with an origin on a closed link at departure time but the vehicles which have a safe origin but soon-to-be or closed downstream path which prevents them from reaching their destinations.

It is not challenging to notify vehicles on a closed link at the time of departure that they cannot travel and leave their origin because most of the times it is visually obvious. Yet, this framework is highly beneficial in notifying the vehicles which are initially in groups 4 and 5 (vehicles with safe origins) and then moved to group 3, because it is not evident to them that they will be incapable of reaching their destinations due to downstream and approaching floods.
Table 11: Results - Number of vehicles per category (Test 2 a)

<table>
<thead>
<tr>
<th>Category</th>
<th>Concentrated (middle)</th>
<th>Concentrated (side)</th>
<th>Dispersed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unaffected vehicles</td>
<td>4064</td>
<td>2890</td>
<td>4729</td>
</tr>
<tr>
<td>Number of affected vehicles</td>
<td>938</td>
<td>2112</td>
<td>273</td>
</tr>
<tr>
<td>Number of routed vehicles</td>
<td>568</td>
<td>1458</td>
<td>234</td>
</tr>
</tbody>
</table>

Table 12: Results - Number of vehicles per group (Test 2 a)

<table>
<thead>
<tr>
<th>Group</th>
<th>Initial grouping</th>
<th>Final grouping</th>
<th>Initial grouping</th>
<th>Final grouping</th>
<th>Initial grouping</th>
<th>Final grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group1</td>
<td>55</td>
<td>55</td>
<td>66</td>
<td>66</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Group2</td>
<td>27</td>
<td>27</td>
<td>77</td>
<td>77</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Group3</td>
<td>344</td>
<td>370</td>
<td>470</td>
<td>654</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>Group4</td>
<td>421</td>
<td>410</td>
<td>741</td>
<td>738</td>
<td>154</td>
<td>154</td>
</tr>
<tr>
<td>Group5</td>
<td>91</td>
<td>76</td>
<td>758</td>
<td>577</td>
<td>67</td>
<td>67</td>
</tr>
</tbody>
</table>

Table 13 shows the basic statistical measures of the computation times of the routed vehicles for each scenario and Table 14 shows the average computation times per group per scenario. Although scenario 2 resulted in the highest number of affected vehicles, it recorded the lowest average computation time (0.99 seconds) and the lowest minimum computation time (0.30 seconds). Thus, the computation time and the impact of the flood do not vary in the same direction. In scenario 3, a maximum computation time of 51.28 seconds is recorded for a vehicle in group 4 (vehicles with soon-to-be flooded or closed destinations). This is the same vehicle (ID: 4394) that is previously discussed in Test 1 in scenario 1 and which needed significant computation time just for the search for a new safe destination. Note that the maximum computation times recorded in scenario 1 (concentrated middle) and scenario 2 (concentrated side) also belong to vehicles in group 4. Consequently, in future connected vehicle environments, it is important to consider limiting the search for a new destination to a certain duration threshold and prompting the users to enter final destinations of their choice to enhance the computation times of vehicles in groups 1 and 4 (vehicles with soon-to-be flooded or closed destinations). According to Table 14, vehicles in group 2 resulted in the highest average computation time in scenario 3 (dispersed flood). While the group with highest average computation times in
scenario 1 and 2 (concentrated floods) is group 1. Groups 1 and 2 both require the generation of two hyperpaths; the first from the origin to a first stop in buffer 1 and the second from the node in buffer 1 to the final destination. Vehicles in group 1 (vehicles with soon-to-be flooded origins and soon-to-be flooded or closed destinations) have to be assigned to new safe destinations but have to make a stop first at a node in buffer 1 to ensure that they exit the danger zone as soon as possible. The computation time required to route vehicles in group 1 can be broken into 4 parts: 1) time required to search for the closest node in buffer 1 (first stop), 2) time required to compute the hyperpath from the origin to buffer 1, 3) time required to find the closest node in buffer 2 (destination), 4) time required to compute the hyperpath from buffer 1 to buffer 2. The computation time required to route vehicles in group 2 can be broken into three parts: 1) time required to search for the closest node in buffer 1 (first stop), 2) time required to compute the hyperpath from the origin to buffer 1, 3) time required to compute the hyperpath from buffer 1 to the final destination. Consequently, to minimize the average computation time for vehicles in group 1 or 2 in future connected vehicle environment implementation, a vehicle in group 1 and 2 should be provided with the first hyperpath once it is generated. This way the vehicle can initiate its trip while the second hyperpath is being computed.

Furthermore, the group with the lowest average computation time is group 5. This is expected because its computation time only comprises the time to find a new path from the original origin to the destination of the vehicle. Regarding group 4, only one path has to be computed but the computation time for routing these vehicles with soon-to-be flooded or closed destinations comprises also the time to search for a new safe destination. Consequently, and as discussed earlier, the computation time of vehicles in this group can be enhanced by stopping the search after scanning a specific number of tentative destinations.
Table 13: Results – Computation times for different flood locations (Test 2 a)

<table>
<thead>
<tr>
<th>Computation times (seconds)</th>
<th>Location</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Concentrated (middle)</td>
<td>Concentrated (side)</td>
</tr>
<tr>
<td>Average</td>
<td>1.56</td>
<td>0.99</td>
</tr>
<tr>
<td>Min</td>
<td>0.35</td>
<td>0.30</td>
</tr>
<tr>
<td>Max</td>
<td>6.24</td>
<td>22.06</td>
</tr>
</tbody>
</table>

Table 14: Results - Average computation times per group for different flood locations (Test 2 a)

<table>
<thead>
<tr>
<th>Average computation times (in seconds)</th>
<th>Location</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Concentrated (middle)</td>
<td>Concentrated (side)</td>
</tr>
<tr>
<td>Group 1</td>
<td>3.18</td>
<td>1.78</td>
</tr>
<tr>
<td>Group 2</td>
<td>2.04</td>
<td>1.20</td>
</tr>
<tr>
<td>Group 4</td>
<td>1.86</td>
<td>1.31</td>
</tr>
<tr>
<td>Group 5</td>
<td>0.96</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Figure 20 indicates the percentage of routed vehicles within each computation time interval per scenario. The change in the flood’s location and the higher impact that a concentrated flood has on the network did not result in higher computation times. For all scenarios, the computation interval that is recorded by the highest percentage of routed vehicles is 0.5 to 1 seconds. Consequently, when varying the flood location, the computation time is insensitive to the variation in location of the flood. If the impact of the flood is higher, the computation time is not necessarily higher or lower and hence no trend is observed.
Figure 20: Results - Percentage of routed vehicles within each computation time interval per scenario (Test 2 a)
Test 2 – Part b: Sensitivity of the results to various flood depths

In this study, we assume that locations at lower elevations are more prone to floods. To generate a flood scenario, we assume that the links that are expected to flood are the links located at an elevation less than or equal to X m. This test consists of evaluating the sensitivity of the framework’s output to different flood depths or severity (i.e. different elevation thresholds X). The following scenarios are tested:

- Scenario 1: Links at 1.7 m elevation and below are expected to flood
- Scenario 2: Links at 2 m elevation and below are expected to flood (base scenario)

The reason why a 1.7 m elevation threshold is selected in scenario 1 is because the number of links at this elevation and below is 94. This number is almost half of the number of flooded links in scenario 2 (the number of flooded links less than or equal to 2 m elevation is 178). Note that the number of links at an elevation less than or equal to 1.6 m is only 61.

RESULTS

As shown in Table 15, as the flood depth increases, the number of affected vehicles and routed vehicles (vehicles in group 1, 2, 4 and 5) increase. This result was expected because with increased flood depth, more locations will flood, consequently affecting additional vehicles. The percent increase in the number of affected vehicles is slightly higher than the one for routed vehicles. This is due to the increase in the number of vehicles in group 3 which do not receive routing assistance as they are required to wait at the origin.

Furthermore, according to Table 16, no vehicle belongs to group 2 (vehicles with an origin on soon-to-be flooded link at departure time) in scenario 1. Yet, in scenario 2, with the additional flooded links, 10 vehicles are added to group 2 and this is due to the fact that new origins are on soon-to-be flooded or closed links. Similarly, the number of vehicles in groups 3 (vehicles with closed origins) and group 4 (vehicles with soon-to-be flooded or closed destinations) increased after the increase of flood depth. However, after increasing the flood elevation threshold, no change in the number of vehicles in group 5 (vehicles with a soon-to-be flooded or closed intermediate link) is noted. This does not mean that there is no affected vehicle with at least one intermediate soon-to-be flooded or closed link at an elevation greater than 1.7 m and less than or equal to 2 m. Due to the group prioritization discussed in Section 3.4.2., a vehicle that has one
intermediate soon-to-be flooded or closed link but that satisfies the conditions of one of the groups 1, 2, 3 and 4 is not listed in group 5.

Table 15: Results - Number of vehicles per category for different flood depths (Test 2 b)

<table>
<thead>
<tr>
<th>Category</th>
<th>Scenario</th>
<th>1</th>
<th>2</th>
<th>% change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unaffected vehicles (Group 6)</td>
<td></td>
<td>4962</td>
<td>4891</td>
<td>-1.43</td>
</tr>
<tr>
<td>Number of affected vehicles (Group 1, 2, 3, 4 and 5)</td>
<td></td>
<td>40</td>
<td>111</td>
<td>177.5</td>
</tr>
<tr>
<td>Number of routed vehicles (Group 1, 2, 4, and 5)</td>
<td></td>
<td>39</td>
<td>100</td>
<td>156.41</td>
</tr>
</tbody>
</table>

Table 16: Results - Number of vehicles per group for different flood depths (Test 2 b)

<table>
<thead>
<tr>
<th>Group</th>
<th>Scenario</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group1</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Group2</td>
<td></td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Group3</td>
<td></td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Group4</td>
<td></td>
<td>12</td>
<td>63</td>
</tr>
<tr>
<td>Group5</td>
<td></td>
<td>27</td>
<td>27</td>
</tr>
</tbody>
</table>

On the other hand, as shown in Table 17, with the increase in flood depth, the average computation time increased from 0.7 seconds in scenario 1 to 1.47 seconds in scenario 2. Similarly, the minimum computation time slightly increased from 0.35 in scenario 1 to 0.39 seconds in scenario 2. In addition, the maximum computation time recorded in scenario 1 is 1.66 seconds while the maximum computation time in scenario 2 is 49.04 seconds. The vehicle that recorded this maximum computation time in scenario 2 is a vehicle in group 4 that needs to be assigned to a new reachable and safe destination (vehicles with maximum computation time in test 1 network 1, (ID: 4394) discussed earlier). Yet, this vehicle is not affected in scenario 1 since its destination is at an elevation above 1.7 m and hence is not considered flooded in scenario 1. Similarly, as shown in Table 18, the average computation time of group 4 increases with the increase in flood depth due to the additional computational time required for the search for a new accessible and safe destination as the number of flooded links increases. However, the average computation time of group 5 negligibly decreased. Yet, the reason behind this decrease is not
necessarily due to the change in the flood depth. Slight variation in the computation time for vehicles in this group is possible because the computation time of this group only comprises the time to generate a hyperpath from the origin to the destination and is subject to vary with the distance that separates the origin and destination. In other words, when the origin and destination of a given vehicles are close in space, less links are scanned and added to the hyperpath, thus resulting in smaller computation times.

Table 17: Results – Computation times for different flood depths (Test 2 b)

<table>
<thead>
<tr>
<th>Computation times (seconds)</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Average</td>
<td>0.70</td>
</tr>
<tr>
<td>Min</td>
<td>0.35</td>
</tr>
<tr>
<td>Max</td>
<td>1.66</td>
</tr>
</tbody>
</table>

Table 18: Results - Average computation times per group for different flood depths (Test 2 b)

<table>
<thead>
<tr>
<th>Average computation times (in seconds)</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Group 2</td>
<td></td>
</tr>
<tr>
<td>Group 4</td>
<td>0.79</td>
</tr>
<tr>
<td>Group 5</td>
<td>0.66</td>
</tr>
</tbody>
</table>

As shown in Figure 19, in scenario 1, all recorded computation times for routed vehicles are computed within 2 seconds and the highest percentage of computation times lies between 0.5 and 1 second (57%). In scenario 2, the same computational time interval recorded the highest percentage (75%); yet, 3 vehicles from group 4 resulted in a computation time greater than 2 seconds because the computation time includes the time to find the new destination which is extensively time-consuming.

Therefore, the increase in the number of flooded links (increase in flood depth) results in an increase in the computation times of the routed vehicles. Thus, further flood depth increase should be tested in the future to check if larger floods will impose some limitations to the framework.
With lower flood depths (scenario 1), the framework requires less time to generate the routing assistance for vehicles that can leave their origin. Consequently, the required computational power of in-vehicle navigation systems is the one needed during the event of high flood depths. During the generation of the routing assistance, vehicles are expected to be waiting at their corresponding origins. During this wait time, vehicles can be notified of the locations to avoid in case of any deviation.

Figure 21: Percentage of routed vehicles within each computation time interval for different flood depths
Test 2 – Part c: Sensitivity of the results to various flood intensities

In this test, the sensitivity of the framework’s results to different flood intensities, in terms of speed, is evaluated. The more intense the flood is, the earlier it occurs and the earlier the roads are considered soon-to-be flooded or closed. A faster flood progress is expected to have a larger impact on the road users.

The following scenarios are tested and illustrated in Figure 22:

- Scenario 1: Links at elevation below 2 m flood at time $t=20$ min

  Table 19: Flood time ($t_4$) per elevation interval (Scenario 1)

<table>
<thead>
<tr>
<th>Elevation interval</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation $\leq 0.5$ m</td>
<td>20</td>
</tr>
<tr>
<td>$0.5 \text{ m} &lt; $ Elevation $\leq 1$ m</td>
<td>20</td>
</tr>
<tr>
<td>$1 \text{ m} &lt; $ Elevation $\leq 1.5$ m</td>
<td>20</td>
</tr>
<tr>
<td>$1.5 \text{ m} &lt; $ Elevation $\leq 2$ m</td>
<td>20</td>
</tr>
</tbody>
</table>

- Scenario 2: The links’ times of flooding are dispersed (early dispersion) (base scenario)

  Table 20: Flood time ($t_4$) per elevation interval (Scenario 2)

<table>
<thead>
<tr>
<th>Elevation interval</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation $\leq 0.5$ m</td>
<td>20</td>
</tr>
<tr>
<td>$0.5 \text{ m} &lt; $ Elevation $\leq 1$ m</td>
<td>30</td>
</tr>
<tr>
<td>$1 \text{ m} &lt; $ Elevation $\leq 1.5$ m</td>
<td>40</td>
</tr>
<tr>
<td>$1.5 \text{ m} &lt; $ Elevation $\leq 2$ m</td>
<td>50</td>
</tr>
</tbody>
</table>

- Scenario 3: The links’ times of flooding are dispersed (late dispersion)

  Table 21: Flood time ($t_4$) per elevation interval (Scenario 3)

<table>
<thead>
<tr>
<th>Elevation interval</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation $\leq 0.5$ m</td>
<td>20</td>
</tr>
<tr>
<td>$0.5 \text{ m} &lt; $ Elevation $\leq 1$ m</td>
<td>40</td>
</tr>
<tr>
<td>$1 \text{ m} &lt; $ Elevation $\leq 1.5$ m</td>
<td>60</td>
</tr>
<tr>
<td>$1.5 \text{ m} &lt; $ Elevation $\leq 2$ m</td>
<td>60</td>
</tr>
</tbody>
</table>
Figure 22: Time of flood (t4) distribution in each scenario

The three scenarios listed above have the same set of flooded links (links at elevation 2 m or below). However, the times of flood (t4) differ. In scenario 1, all of these links are assumed to flood at time t4=20 minutes. In scenario 2, the times of flood are dispersed and the flood first appears at time t=20 min for the links at the lowest elevation. In scenario 3, the times of flood are more time-spaced than the ones in scenario 2. It is important to note that, in scenario 3, even though the flood (t4) of the links at an elevation above 1 m occurs at the end of the one-hour simulation, the flood on these links has an impact on the road users since the network starts being affected by the flood on these links at time = 45 min (= t4 – empty link interval – minimum clearance and additional clearance interval), after which vehicles in groups 4 and 5 and vehicles in groups 1 and 2 that reached a safe node are prevented from traveling along the soon-to-be flooded links.

RESULTS

As shown in Table 22, the highest number of affected vehicles is recorded in scenario 1 (all at 20 min), followed by scenario 2 then scenario 3. Therefore, as the flood is more spaced in time, less vehicles are affected. When the flood is dispersed in time (scenario 2 and 3), 90% (=100/111=53/59) of the affected vehicles are routed while in scenario 1 (flood at time t=20 min), 86% (=234/273) of the affected vehicles were routed. So, as flood appears suddenly and all at once, more vehicles are expected to wait at their origins.
Scenario 1 can represent conditions during severe floods since the floods on links occur earlier in time and hence last longer. In addition, this scenario can also represent cases in which there is a lack of enough information to provide accurate estimates of the times of flood (t4) and in which links are subsequently closed earlier in time for safety measures. As shown in Table 23, when shifting from scenario 1 to scenario 2 and then to scenario 3, the number of vehicles per group decreases but this is directly linked to the decrease in the number of affected vehicles.

In scenario 1, one vehicle is added to group 1. The same vehicle is considered unaffected in scenarios 2 and 3. In scenario 1, the vehicle is considered affected due to the fact that with earlier flood time (in scenario 1), the origin and destination are located on soon-to-be flooded links. Thus, the vehicle’s intended path is adjusted so that the vehicle is first routed to a safe stop in buffer 1 and next to a new destination in buffer 2. The percentages of affected vehicles in group 2 in scenarios 2 and 3 (dispersed) are higher than the one in scenario 1 (flood at time= 20 min) while the percentages of affected vehicles in group 3 in scenarios 2 and 3 are lower than the one in scenario 1. In fact, a vehicle that is in group 3 in scenario 1 because its origin is closed at departure time, will either remain in group 3, be added to group 2 or be considered unaffected in scenarios 2 and 3 (more dispersed and late flood). Therefore, it is expected to observe an increase in the percentage of affected vehicles in group 2 along with a decrease in the percentage of affected vehicles in group 3. The percentages of affected vehicles in group 4 and 5 were approximately identical across scenarios; between 56 and 59% for group 4 and 22 and 24% for group 5. These vehicles are considered affected if an intermediate link or destination is soon-to-be flooded or closed at their expected time of entry or arrival respectively. If their corresponding expected time of entry/arrival occurs after the latest time of flood of the link, the vehicle is not affected and thus less sensitive to varying flood intensities.
Table 22: Results - Number of vehicles per category for different flood intensities (Test 2 c)

<table>
<thead>
<tr>
<th>Category</th>
<th>Scenario</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unaffected vehicles (Group 6)</td>
<td></td>
<td>4729</td>
<td>4891</td>
<td>4943</td>
</tr>
<tr>
<td>Number of affected vehicles (Group 1,2,3,4 and 5)</td>
<td></td>
<td>273</td>
<td>111</td>
<td>59</td>
</tr>
<tr>
<td>Number of routed vehicles (Group 1,2,4 and 5)</td>
<td></td>
<td>234</td>
<td>100</td>
<td>53</td>
</tr>
</tbody>
</table>

Table 23: Results - Number of vehicles and percentage of affected vehicles per group for different flood intensities (Test 2 c)

<table>
<thead>
<tr>
<th>Group</th>
<th>Scenario</th>
<th>Number</th>
<th>%</th>
<th>Number</th>
<th>%</th>
<th>Number</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group1</td>
<td>1</td>
<td>1</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Group2</td>
<td>2</td>
<td>12</td>
<td>4.4</td>
<td>10</td>
<td>9</td>
<td>5</td>
<td>8.5</td>
</tr>
<tr>
<td>Group3</td>
<td>3</td>
<td>39</td>
<td>14.3</td>
<td>11</td>
<td>10</td>
<td>6</td>
<td>10.2</td>
</tr>
<tr>
<td>Group4</td>
<td>4</td>
<td>154</td>
<td>56.4</td>
<td>63</td>
<td>56.7</td>
<td>35</td>
<td>59.3</td>
</tr>
<tr>
<td>Group5</td>
<td>5</td>
<td>67</td>
<td>24.5</td>
<td>27</td>
<td>24.3</td>
<td>13</td>
<td>22.0</td>
</tr>
</tbody>
</table>

Table 24 shows the basic statistical measures of the computation times of the routed vehicles in each scenario. A decrease in the maximum computation time is noted as going from scenario 1 to scenario 3. The maximum computation time recorded in each scenario is for the same vehicle (ID: 4394) that belongs to group 4 and which is discussed previously in Test 1. The recorded maximum computation time decreases with decreasing flood intensity. Thus, as intensity decreases, it takes less time to find a new safe destination that can be reached by this vehicle. An insignificant variation in the recorded minimum computation times across scenarios is noted. Nevertheless, the average computation time increased as the flood timings is more dispersed in time. One reason leading to this observation is that as floods happen late in time (as going from scenario 1 to 2 to 3), the vehicles with long trips are the ones that remain affected while the ones with short trips are the first to become unaffected because they will be capable of completing their trips before any link in their path becomes soon-to-be flooded or closed. So with more time-dispersed floods (less intense), the vehicles with longer trips are the ones
affected and routed and the computation of the hyperpath for these vehicles with long trips takes more time but is still considered small.

Table 24: Results – Computation times for different flood locations (Test 2 c)

<table>
<thead>
<tr>
<th>Computation times (seconds)</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1.08</td>
<td>1.47</td>
<td>2.16</td>
</tr>
<tr>
<td>Min</td>
<td>0.38</td>
<td>0.39</td>
<td>0.38</td>
</tr>
<tr>
<td>Max</td>
<td>51.28</td>
<td>49.04</td>
<td>47.89</td>
</tr>
</tbody>
</table>

As shown in Figure 23, for all flood timing scenarios, more than 60% of the routed vehicles needed a computation time between 0.5 and 1 second. In scenario 1 which is considered to be the most intense flood, the highest percentage of routed vehicles within 2 seconds (98% = 8+82+6+2) is noted. Thus, intense floods (flood happening earlier in time) do not impose any impact on the computation times.

![Figure 23: Percentage of routed vehicles within each computation time interval for different flood intensities](image-url)
Summary: Test 2

In the future connected vehicle environment, weather and flood forecast systems play a major role in defining the flood characteristics and subsequently in assessing its impact, which highly depends on the flood location, severity and speed. According to the results obtained in Test 2-a, with different flood locations, the transportation network is affected differently. In our case, a flood that occurs in concentrated areas has more impact on the network than a flood that is dispersed across the network. Yet, this should not be considered as a general rule since the impact of a flood highly depends on the relative location of the origins and destinations of the vehicles in the system with respect to the flooded links. Furthermore, according to Test 2-b, when the severity of a specific flood increases, more links are expected to flood, and hence additional vehicles are affected. This is why it is important to accurately estimate the flood’s severity which dictates the number of flooded links. Overestimation of the severity of a flood will result in more affected vehicles and in more vehicles unable to complete their original trip while underestimation has safety implications. Similarly, according to Test 2-c, as the flood on links occurs earlier in time, additional vehicles (whose trip was intact with late flood) become affected. If accurate weather and flood forecasts are unavailable, links may be closed ahead of time for safety purposes. In other words, the expected time of flooding (t4) of each link could be set at a time before the real t4, unnecessarily affecting more vehicles that could have completed their original trip safely.

In all of these tests, only one-hour of simulated traffic is tested and it represents the phase during which the flood starts to appear. In real life, the flood reaches a point where no further increase in depth and intensity occurs (i.e. no additional link is flooded). After this point, the locations that are flooded remain the same. Hence, no vehicle will be added to group 1 and 2 because there is no link that is considered soon-to-be flooded. The vehicles will either be unaffected or will be affected due to a closed link in their path (vehicles categorized under group 3, 4, 5 or 6).
Test 3 – Part a: The sensitivity of the results to various frequencies of weather and flood forecast updates

In this study, we assume that the TMC updates and receives the flood input data from weather and flood forecasts systems at a specific frequency. In this test, we evaluate the results’ sensitivity to variation in the weather and flood forecasts’ update frequency.

The following scenarios are tested and represented in Table 25:

- Scenario 1: 1-minute update frequency (or near-continuous update)
- Scenario 2: 5-minutes update frequency
- Scenario 3: 10-minutes update frequency (base scenario)

**Table 25: Time of flood (t4) corresponding to each elevation interval in each scenario**

<table>
<thead>
<tr>
<th>Elevation interval</th>
<th>Real time of flood</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>t4</td>
<td>t4</td>
</tr>
<tr>
<td>Elevation &lt;= 0.5 m</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>0.5 m&lt; Elevation &lt;=1 m</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>1 m&lt; Elevation &lt;=1.5 m</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>1.5 m&lt; Elevation &lt;=2 m</td>
<td>59</td>
<td>59</td>
</tr>
</tbody>
</table>

According to Table 25, as the weather and flood forecasts are updated in the TMC database less frequently, early closures result. For instance, if continuous updates (i.e. every 1-minute) indicates that a given link floods at time t = 28 min, the time of flood (t4) of the same link is considered at time t= 25 min and at time t=20 min with an update frequency of 5 minutes and 10 minutes, respectively.
RESULTS

A safety margin is already provided by preventing all vehicles from traveling a link during the empty link interval (at time t>=t3) and by prohibiting certain groups of vehicles from entering a link during the minimum clearance and additional clearance interval (at time t>=t1). Thus, less frequent updates could result in closure of the links at an additional earlier time with no benefit to the system, compared to more frequent updates. For example, if the true time of the flood (t4) is 28 min, this is the time used for near-continuous updates, while if 10-minute updates are used, time of flood (t4) is set at time t = 20 and the links are considered closed and soon-to-be flooded 8 minutes earlier than the actual flood time.

As shown in Table 26, with less frequent weather and flood forecast updates, the numbers of affected vehicles and routed vehicles (i.e. vehicles in groups 1, 2, 4 and 5) increase. Links are closed and considered as soon-to-be flooded earlier in time. Hence, fewer weather and flood forecast updates result in a lengthier impact on the road users. In scenario 1, the first set of links considered as soon-to-be flooded are at time t= 13 min (=28 – Empty link interval of 5 min – Minimum clearance and additional clearance interval of 10 min) while the first set of links considered as soon-to-be flooded in scenario 2 is at time t=10 minutes and at time t=5 minutes in scenario 3.

As shown in Table 27, fewer updates led to the increase of the number of vehicles in each group for all cases except the shift from a 1-minute frequency to a 5-minute frequency in which the number of vehicles in group 3 did not change. For instance, fewer updates result in earlier “t3” for all links. Let durations X_i be the difference between the time t3 of a link in a 1-minute frequency updates and the time t3 the link in a 5-minute frequency updates, “i” being the link’s elevation interval. Based on the simulated vehicle data obtained from VISSIM, no additional vehicle has a departure on one of the flooded links during this duration X_i; thus, no additional vehicle is added to group 3 when shifting from scenario 1 to scenario 2. However, when shifting from a 5-minute frequency to a 10-minute frequency, four additional vehicles are added to group 3 and are subsequently prohibited to leave their origins because no hyperpath to their final destination is found. These vehicles belong to group 2 in scenarios 1 and 2 during which they were able to leave their soon-to-be flooded origins to reach their corresponding destinations.
Table 26: Results - Number of vehicles per category for different update frequencies (Test 3 a)

<table>
<thead>
<tr>
<th>Category</th>
<th>Update Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 min</td>
</tr>
<tr>
<td>Number of unaffected vehicles</td>
<td>4939</td>
</tr>
<tr>
<td>Number of affected vehicles</td>
<td>63</td>
</tr>
<tr>
<td>Number of routed vehicles</td>
<td>56</td>
</tr>
</tbody>
</table>

Table 27: Results - Number of vehicles per group for different update frequencies (Test 3 a)

<table>
<thead>
<tr>
<th>Group</th>
<th>Update Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 min</td>
</tr>
<tr>
<td>Group1</td>
<td></td>
</tr>
<tr>
<td>Group2</td>
<td>5</td>
</tr>
<tr>
<td>Group3</td>
<td>7</td>
</tr>
<tr>
<td>Group4</td>
<td>37</td>
</tr>
<tr>
<td>Group5</td>
<td>14</td>
</tr>
</tbody>
</table>

In Table 28, the basic statistical measures of the computation times of the routed vehicles for each scenario are shown. According to the results, the average computation time decreases with fewer updates. The reason behind this is similar to the reason why the computation time increases with more time-dispersed times of flood (t4) in Test 2-c. With more updates, the vehicles with longer trips remain affected and these vehicles require more computation time to generate their corresponding path than the ones with short trips. The maximum recorded computation time in all the scenarios belongs to the same vehicle (ID: 4394) that was discussed earlier.

Table 28: Results – Computation times for different flood locations (Test 3 a)

<table>
<thead>
<tr>
<th>Computation times (sec)</th>
<th>Update Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 min</td>
</tr>
<tr>
<td>Average</td>
<td>2.09</td>
</tr>
<tr>
<td>Min</td>
<td>0.36</td>
</tr>
<tr>
<td>Max</td>
<td>49.03</td>
</tr>
</tbody>
</table>

Figure 24 shows the percentages of routed vehicles that fall within each computation time interval for different update frequencies. The percentage of routed vehicles that have a computation time greater than or equal to 0.5 second and less than 1 second has the highest
percentage at all update frequencies. For instance, nearly identical percentages are reported for each computation time interval which indicates that the variation in update frequency does not imply any impact on the computation time but it only affects the impact on the system in terms of the number of affected vehicles. Note that all the vehicles that have a computation time greater than or equal to 2 seconds belong to group 4. Since these vehicles cannot reach their original destinations, a new destination that is a reachable node in buffer 2 and that is the closest to the original destination is assigned for each vehicle. It is the search for a new destination that is time-consuming but which can be limited in future works.

![Figure 24: Percentage of routed vehicles within each computation time interval for different forecast update frequencies](image-url)
Test 3 – Part b: The sensitivity of the results to various empty link intervals

As a safety margin, an empty link interval is defined by policy makers to prohibit all movement of vehicles for a specific time period before the expected time of flood \( t_4 \). All groups of vehicles are not allowed to enter nor exit the links after their corresponding times \( t_3 \). The empty link interval is a practical variable that depends on the traffic and weather conditions. For demonstration purposes, the empty link interval, in this study, is considered fixed for all links. In this test, the sensitivity of the framework’s results to different empty link interval lengths is evaluated.

The following scenarios are tested and are represented in Table 29:

- Scenario 1: Empty link interval = 5 min (base scenario)
- Scenario 2: Empty link interval = 10 min

Table 29: Time \( (t_3) \) corresponding to each elevation interval in each scenario

<table>
<thead>
<tr>
<th>Elevation interval</th>
<th>Empty link interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 min</td>
</tr>
<tr>
<td>Elevation &lt;= 0.5 m</td>
<td>t_4</td>
</tr>
<tr>
<td>0.5 m&lt; Elevation &lt;=1 m</td>
<td>20</td>
</tr>
<tr>
<td>1 m&lt; Elevation &lt;=1.5 m</td>
<td>30</td>
</tr>
<tr>
<td>1.5 m&lt; Elevation &lt;=2 m</td>
<td>40</td>
</tr>
</tbody>
</table>

RESULTS

As shown in Table 30, the increase in the empty link interval duration from 5 minutes to 10 minutes led to the increase in the number of affected and routed vehicles. This result is expected since vehicles are prohibited to enter the links at an earlier time. In other words, if the empty interval is equal to 10 minutes and if a link is expected to flood at time \( t_4=30 \) min, all movements are prohibited after time \( t=20 \) min. However, if the empty link interval is equal to 5 minutes, all vehicle movements are prevented after time \( t=25 \) min for the same link and time of flood \( t_4 \). For larger empty link interval, the links start being considered as soon-to-be flooded at an earlier time.
According to results in Table 31, as expected, the number of vehicles per group increased with the increase in the empty interval length except in group 2. The reason why a decrease is noticed in group 2 is because two vehicles shifted from group 2 to group 3 because their corresponding origins which are located on soon-to-be flooded links at their times of departure with an empty link interval of 5 min are now located on closed links with an empty link interval of 10 min.

Table 30: Results - Number of vehicles per category for different empty link interval (Test 3 b)

<table>
<thead>
<tr>
<th>Category</th>
<th>Empty link interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 min</td>
</tr>
<tr>
<td>Number of unaffected vehicles</td>
<td>4891</td>
</tr>
<tr>
<td>Number of affected vehicles</td>
<td>111</td>
</tr>
<tr>
<td>Number of routed vehicles</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 31: Results - Number of vehicles per group for different empty link interval (Test 3 b)

<table>
<thead>
<tr>
<th>Group</th>
<th>Empty link interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 min</td>
</tr>
<tr>
<td>Group1</td>
<td>0</td>
</tr>
<tr>
<td>Group2</td>
<td>10</td>
</tr>
<tr>
<td>Group3</td>
<td>11</td>
</tr>
<tr>
<td>Group4</td>
<td>63</td>
</tr>
<tr>
<td>Group5</td>
<td>27</td>
</tr>
</tbody>
</table>

Table 32 shows the basic statistical measures of the computational times of routed vehicles. According to the results, the average computation time decreased as the empty time interval increased and this can be due to more vehicles with lengthier trips being routed as the empty interval decreases. Yet, the difference is still small and does not impose any limitation to the framework. Regarding the maximum recorded computation time in each scenario, it is the same vehicle (ID: 4394) that also resulted time-consuming search for a new destination. As for the minimum recorded computation time per scenario, the difference is negligible.
Table 32: Results – Computation times for different empty link intervals (Test 3 b)

<table>
<thead>
<tr>
<th>Empty link interval</th>
<th>5 min</th>
<th>10 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation times (in sec)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1.47</td>
<td>1.31</td>
</tr>
<tr>
<td>Min</td>
<td>0.39</td>
<td>0.38</td>
</tr>
<tr>
<td>Max</td>
<td>49.04</td>
<td>48.44</td>
</tr>
</tbody>
</table>

Figure 25 shows the percent of vehicles per computation time range per scenario. The difference in terms of computation times between the two scenarios is small. Nearly identical trends are observed across scenarios; thus the percent of routed vehicles in each computation time interval is insensitive to variations in the duration of the empty link interval.

Figure 25: Percentage of routed vehicles within each computation time interval for different empty link intervals
Test 3 – Part c: The sensitivity of the results to various additional clearance intervals

Similar to the empty link interval, the minimum clearance and additional clearance interval acts as a safety margin but only prevents vehicles located outside the soon-to-be flooded zone to enter and travel along the soon-to-be flooded links. As discussed in section 3.3.3., the minimum clearance interval depends on the link characteristics and on the weather and traffic conditions. In this study, the minimum clearance interval is considered fixed for all links and equal to 5 minutes. The additional clearance interval is a practical variable that is defined by policy makers or a transportation agency. It acts as an optional extension to the minimum clearance time and can be considered equal to zero when the minimum clearance interval is sufficient to clear the soon-to-be flooded links.

This test consists of evaluating the sensitivity of the framework’s results to different additional clearance interval durations. The following scenarios are tested:

- Scenario 1: Additional clearance interval = 0 min
- Scenario 2: Additional clearance interval = 5 min (base scenario)
- Scenario 3: Additional clearance interval = 10 min

RESULTS

As shown in Table 33, the number of affected vehicles and routed vehicles increase with the increase in the duration of the additional clearance interval. This result is highly expected as links are considered soon-to-be flooded earlier in time. According to Table 34, an increase in the number of vehicles per group is noted with lengthier additional clearance intervals; a result that is also expected except for group 3. In fact, with additional minimum clearance time, the interval during which links are considered soon-to-be flooded are lengthened. Hence, no vehicles with an origin on a closed link should be added to group 3 with the increase in the additional minimum clearance interval. The same number of vehicles in group 3 is reported in scenario 1 and scenario 2; but with further increase in the additional minimum clearance (from scenario 2 to scenario 3), one additional vehicles is added. After analyzing the results, it turned out that this vehicle does not have an origin on a closed link but is a vehicle whose hyperpath to a safe destination does not
exist. For instance, this vehicle was added to group 3 after execution of the “process” step in the framework.

Table 33: Results - Number of vehicles per category for different additional clearance intervals (Test 3 c)

<table>
<thead>
<tr>
<th>Category</th>
<th>Additional clearance interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 min</td>
</tr>
<tr>
<td>Number of unaffected vehicles</td>
<td>4916</td>
</tr>
<tr>
<td>Number of affected vehicles</td>
<td>86</td>
</tr>
<tr>
<td>Number of routed vehicles</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 34: Results - Number of vehicles and percentage of affected vehicles per group for different additional clearance intervals (Test 3 c)

<table>
<thead>
<tr>
<th>Group</th>
<th>Additional clearance interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 min</td>
</tr>
<tr>
<td>Group1</td>
<td>0</td>
</tr>
<tr>
<td>Group2</td>
<td>5</td>
</tr>
<tr>
<td>Group3</td>
<td>11</td>
</tr>
<tr>
<td>Group4</td>
<td>51</td>
</tr>
<tr>
<td>Group5</td>
<td>19</td>
</tr>
</tbody>
</table>

Tables 35 shows the basic statistical measures of the computational times of routed vehicles and Figure 23 shows the percent of vehicles per computation time range per scenario. According to Table 35, an increase in the average computation time is observed with the increase on the additional clearance interval. In fact, the increase observed between scenario 1 and scenario 2 is mainly due to the jump in the maximum recorded computation time from 1.53 to 49.04 seconds (more time-consuming search for a destination for vehicles in group 4 in scenario 2). Plus, according to Figure 26, in Scenario 1 (null additional clearance interval), all routed vehicles have a computation time less than 2 seconds and the reason behind this is the less time-consuming search for a new destination for vehicles in group 4 with smaller additional clearance intervals.

Note that the vehicles that recorded a computational time greater than or equal to 2 seconds in scenarios 2 and 3 are vehicles which belong to group 4 in scenario 1. These vehicles, in scenario 1, have received routing, yet, the search for a node in buffer 2 to act as the new safe destination is completed faster in scenario 1 than in scenarios 2 and 3 because vehicles, in scenario 1, are
allowed to enter a link that is considered soon-to-be flooded in scenario 2 just due to the
lengthier additional clearance time and not due to earlier time of flood (t4).

The increase in the average computation time observed when shifting from scenario 2 to scenario
3 is due to the higher percentage of routed vehicles with computation time greater than 2 seconds
(3% in scenario 2 compared to 6% in scenario 3). Subsequently, a careful selection of the
additional clearance time is required because it mainly affects the computation time required for
the search for new safe destinations for vehicles with destinations on soon-to-be flooded and
closed links.

Table 35: Results – Computation times for different additional clearance intervals (Test 3 c)

<table>
<thead>
<tr>
<th>Additional clearance interval</th>
<th>0 min</th>
<th>5 min</th>
<th>10 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation times (sec)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.79</td>
<td>1.47</td>
<td>1.80</td>
</tr>
<tr>
<td>Min</td>
<td>0.37</td>
<td>0.39</td>
<td>0.36</td>
</tr>
<tr>
<td>Max</td>
<td>1.53</td>
<td>49.04</td>
<td>52.10</td>
</tr>
</tbody>
</table>

Figure 26: Percentage of routed vehicles within each computation time interval for different additional clearance intervals
Summary: Test 3

The results of Test 3-a, Test 3-b and Test 3-c indicate that the framework’s output depends on policy-related variables and other dependencies such as the frequency of update of the flood and weather forecasts; hence, the selection of these variables is a very delicate task. The frequency of update of the flood and weather forecasts affects the closing time of the links and subsequently dictates the travel impact of the flood. In other words, less frequent updates result in the early closure of links while continuous updates lead to the closures of links as needed (Test 3-a).

Regarding the empty link interval and the additional clearance interval, these two policy-related variables act as a safety margin to prohibit vehicles from travelling the link during a specific interval prior to the time of flood. Increasing the length of these intervals enhances safety; nevertheless, the links will be closed earlier in time and the travel impact of the flood will be more significant (Test 3-b and 3-c).
CHAPTER V: CONCLUSION

The proposed framework assists vehicles that are affected by a flood event. Based on the affected vehicle’s path and departure time and based on the location and time of the flood, the vehicle will either be requested to wait at the origin or will receive routing assistance in the form of a hyperpath. The hyperpath is computed using the hyperstar algorithm developed by Bell [9]. It is a set of alternative paths that connects the origin to the destination of the vehicle while preventing entry to links that are soon-to-be flooded or closed. If the destination of the vehicle is positioned on a link that cannot be entered due to the floods, the vehicle will be assigned to a new safe destination. If the vehicle is blocked and cannot reach any safe destination, it will be instructed to wait at the origin along with the vehicles that are originally positioned on a closed link. Vehicles that are departing from a soon-to-be flooded link will receive a hyperpath to a safe node first to ensure that it exited the flooded link or area as quickly as possible. Then, they will receive a second hyperpath to resume their trip to their final destination.

The proposed framework relies on the connected vehicle environment which allows the communication between a traffic management center and equipped vehicles. Even though the whole process is supposed to be executed within the in-navigation systems, the traffic management center’s role is to gather and process the required information such as link performance measures and weather and flood forecasts and send them to the vehicles in which the process is executed and routing assistance or warnings are generated. The proposed framework, coded in java, is implemented with simulated vehicle data on two transportation networks modeled in VISSIM based on the city of Virginia Beach. The main objectives of the tests are to evaluate the scalability of the framework to different transportation network sizes, and the sensitivity of the results to different flood characteristics and to variation in policy-related variables and other dependencies such as the weather and flood forecasts’ update frequency. In each test, a set of scenarios is created and executed. The framework’s output that is thoroughly analyzed comprises the impact of the flood on the system in terms of affected vehicles, routed vehicles and distribution among groups along with the assessment of the computation times required to route vehicles in each scenario of each test.

First, it is important to consider that the obtained results highly depend on the relative location of the origins and destination of the vehicles with respect to the flooded locations. This
made some observed results, in certain scenarios, network-specific and from which general rules cannot be established. According to the tests, the impact of a flood increased with the increase in flood’s depth and intensity. In addition, concentrated floods resulted in more significant impacts compared to a dispersed one. This is why the collection of accurate weather and flood forecasts are crucial to this framework. Furthermore, the update frequency of the weather and flood systems has implications on the system. With fewer updates, more vehicles are affected. Similarly, the selection of the safety margins (empty link interval and additional minimum clearance) during which vehicles depending on their group are not allowed to enter the link, is critical. Increasing these intervals results in additional safety measures but it also engenders more impact on the system. Therefore, the selection of the best empty link interval and additional clearance interval is necessary to ensure efficiency while also guaranteeing safety of the users.

As for the computation time of routed vehicles, results show that it is sensitive to the size of the network and the number of flooded links (i.e. flood depth). Yet, the obtained computation time for the larger network and the deepest flood are still reasonable. Thus, further increase in the network size and in the flood depth can be tested in the future. According to the obtained results, routed vehicles in group 4 (vehicles with soon-to-be flooded or closed destinations) recorded the highest computation times due to the time-consuming search for new safe destinations. In order to enhance the computation times of these vehicles (and vehicles in group 1 as well), the search for a new destination can be limited and stopped after the evaluation of a maximum number of tentative destinations. Similarly, and to minimize the computation time of vehicles in groups 1 and 2 (vehicles on soon-to-be flooded origins directed to a safe stop first then to the final destination), the user can receive the first hyperpath and start travelling while the second hyperpath is being computed. The computation time of the vehicles is a very important measure since it will dictate the computational power of the in-vehicle navigation systems required by this proposed framework. The computation times recorded from the execution of the tests on a computer that is relatively not powerful are small and will become even smaller if enhancements in the algorithm used to search for new destination are implemented in the future. Hence, using in-navigation systems with the same specifications are enough for this framework. Yet, using more powerful ones would result in even smaller computation times making the proposed framework more and more computationally efficient.
To conclude, this framework is a new tool to be used in the connected vehicle environment. This environment is considered the future platform of transportation systems and all the extensive researches currently and previously conducted made this initially visionary environment, with time, closer to reality. The framework focuses on an aspect of flooding that was not addressed by previous literature. For instance, a flood does not necessarily result in the evacuation of the occupants of the area; yet, in most of the cases with low to moderate severity, it only engenders the entry prevention to flooded roads and thus the rerouting of vehicles. The recorded computation times are small although this is still the first trial and several improvements are already outlined. The proposed framework is a promising and timely tool that is expected to bring, in the future, a lot of benefits in terms of safety and mobility to the roads users exposed to a flood.

Our next step consists of evaluating this framework using a microscopic traffic simulation tool to capture the flow variations due to the assignment of vehicles on the routes. Therefore, expressing the benefits in terms of travel time, fuel consumption and emission and other performance measures will be possible.

In addition, future works include the evaluation of the framework with empty interval and minimum clearance intervals specific for each link instead of assigning the same durations for all links. In fact, these policy-related variables could vary with the different link’s characteristics and weather and flow conditions. Furthermore, with the aim of simulating human behavior, the framework will be extended in a way to allow vehicles that were heading towards a destination on a soon-to-be flooded or closed link, to accept the new safe destination or reject it and select a more preferred safe destination of their choice.

Finally, this framework and more specifically the routing methodology will be revisited so that the hyperstar routing algorithm is triggered even after the trip initiation, at intermediate nodes. Since unexpected events are likely to turn an optimal route into a suboptimal one due to incurred delays, generating an updated hyperpath from downstream intermediate nodes to the destination is a technique that continuously searches for better routes. If significant time savings can be acquired, the new hyperpath from the intermediate node to the destination is displayed to the user who will decide whether he wants to adopt it or not.
REFERENCES

1. Petra Tschakert, B.K., Seth Baum, Chongming Wang, Geographic Perspectives on Sustainability and Human-Environment Systems.


APPENDIX A

Process main class {

Read the input_vehicle file;

For each vehicle v (group, origin, destination, start_time): {
  max_computation_time=0;

  If (group =1) {
    time1(1) = start_time;
    Find the closest node in buffer 2 to destination;
    New destination = closest node in buffer 2;
    Find closest node in buffer 1 to new destination;
    Run partitioning algorithm to the selected node in buffer 1;
    Run hyperstar between origin and node in buffer 1;

    while (hyperpath is empty &amp; next closest node in buffer 1 exists) {
      Find the next closest node in buffer 1 to the destination;
      Run the partitioning algorithm to the selected node in buffer 1;
      Run hyperstar between the origin and the selected node in buffer 1;
    }

    if (hyperpath is empty) {
      add the vehicle to group 3;
    }
    else {
      add the vehicle to group 1;

      time1(1)=end;
      if (time1(1)> max_computation_time)
        max_computation_time=time1(1);

      output = Select_path method (v); 
      add output to ROUTING file;
      Get actual time of arrival at buffer 1;
      If (node in buffer 1 == node in buffer 2)
        Vehicle has reached destination \rightarrow stop;
    }
  }

  If (group =2 ) {
    time2(1) =start_time;
    Find the closest node in buffer 1 to destination;
    Run the partitioning algorithm to the selected node in buffer 1;
    Run hyperstar between the origin and the node in buffer 1;

  }
}

}
while (hyperpath is empty && next closest node in buffer 1 exists) {
    Find the next closest node in buffer 1 to the destination;
    Run the partitioning algorithm to the selected node in buffer 1;
    Run hyperstar between the origin and the selected node in buffer 1;
}

if (hyperpath is empty) {
    add the vehicle to group 3;
}
else {
    add the vehicle to group 2;
    time2(1)=end
    if (time2(1)> max_computation_time)
        max_computation_time=time2;
    output = Select_path method (v)
    add output to ROUTING file
    Get actual time of arrival at buffer 1;
    If (node in buffer 1 == destination)
        Vehicle has reached destination → stop;
    else {
        time2(2)=start;
        set start_time = actual time of arrival at buffer 1;
        Run the partitioning algorithm to the destination;
        Run hyperstar between node in buffer 1 and destination;
        if (hyperpath empty)
            Vehicle is blocked and has to wait at buffer 1; add to subgroup 2b;
        else {
            time2(2)=end;
            if (time2(2)> max_computation_time)
                max_computation_time=time(2);
            output= Select_path method (v);
            add output to ROUTING file;
        }
    }
}

If (group =3) {
    Add the vehicle to group3;
}

If (group 4){
    time4=start_time;
    Find closest node in buffer 2 to destination;
    New destination = closest node in buffer 2;
    Run the partitioning algorithm to the selected node in buffer 2;
    Run hyperstar between origin and the node in buffer 2;
    while (hyperpath empty && next closest node in buffer 2 exists)
        Find the next closest node in buffer to the destination;
        Run the partitioning algorithm to the selected node in buffer 2;
        Run hyperstar between the origin and the selected node in buffer 2;
    }

if (hyperpath is empty) {
    add the vehicle to group 3;
}
else {
    add the vehicle to group 4;
    time4=end;
    if (time4> max_computation_time)
max_computation_time=time1;
output = Select_path method (v);
add output to ROUTING file;

If (group 5) {
    time5=start_time;
    Run partitioning algorithm to destination;
    Run hyperstar between origin and destination;
    if (hyperpath empty)
        add the vehicle to group 3;
    else
        add the vehicle to group 5;
    time5=end;
    if (time5> max_computation_time)
        max_computation_time=time1;
    output = Select_path method (v);
    add output to ROUTING file;
}

If (group 6) {
    Add the vehicle to Outputfile_Group6;
}

Get max computation time
}

select_path method (vehicle) {
    Scan the hyperpath of vehicle v;
    Scan the actual network (with actual data);
    end= origin of v;
    time to destination = start time of v;
    path = empty;
    while(end is not destination){
        find the links in hyperpath exiting node end;
        select a link (generate a random number) and add it to the path;
        time to destination = time to destination + travel time of selected link;
        end= end node of the selected link;
    }
    return array of String (index 0: vehicle id, index1: time of arrival at destination and index 2: path to destination);
}