Black Ice! Using Information Centric Networks for Timely Vehicular Safety Information Dissemination

Jiachen Chen*, Mohammad Jahanian†, and K. K. Ramakrishnan†
* WINLAB, Rutgers University, NJ, USA. Email: jiachen@winlab.rutgers.edu
† University of California, Riverside, CA, USA. Email: mjaha001@ucr.edu, kk@cs.ucr.edu

***Abstract*—Vehicles are increasingly equipped with special purpose sensors and Global Positioning System (GPS) for use in safety applications. Beyond using these sensors, sharing information among vehicles can substantially improve the safety of the overall transportation environment. Enabling each vehicle to get the "right information at the right time", to avoid potentially dangerous situations can be valuable. Information-Centric Networks (ICN) that uses the notion of "named-object" enable information retrieval and delivery regardless of its location, publisher or requester. Using ICN, especially supporting publish/subscribe can provide timely delivery of relevant vehicular safety information. Our V-ICE architecture utilizes Roadside Units (RSUs) to act as infrastructure-based information aggregators to communicate with vehicles that generate notifications of safety-related information. RSUs also disseminate this information to the right vehicles who subscribe to the information relevant to the path they are traveling on. To evaluate the benefit of V-ICE, we demonstrate its use in propagating "black ice" warnings to vehicles that will likely be affected by the black ice event on their routes. The critical need is for the information to be delivered in a timely manner, compared to a server-based infrastructure. This provides other cars sufficient time to react. We build V-ICE's namespace and architecture based on a representative city environment, using the roadways of Luxembourg as an example, and evaluate our approach using a 4-hour traffic trace generated by the SUMO synthetic traffic generator. Our results show that V-ICE scales and performs better than a server-based approach or V2V broadcast, in terms of timeliness, relevance, and network traffic.**

I. Introduction

Vehicular safety has been a long-standing area of concern for transportation systems. It has received more attention recently because of the multitude of sensors and cameras on vehicles, and the ability of vehicles to wirelessly communicate with infrastructure or other vehicles. While many safety improvements such as collision avoidance etc., exploit these sensors and cameras, approaches that exploit both sensor information and communication can further improve vehicular safety. We explore opportunities where a vehicle that experiences a safety related incident can use communication to effectively disseminate information to other vehicles and help them avoid encountering the same incident.

Black ice is "a nearly transparent film of ice on a dark surface that is difficult to see". This phenomena on roads, especially highways with a large number of vehicles moving at high speed, can lead to accidents and (sometimes a cascade of) collisions resulting in injuries and possibly death. Since it is very hard to see, the use of on-vehicle sensors to detect black ice is necessary. However, becoming aware of the existence of black ice by one's own sensor when driving over it is neither sufficient nor even helpful. By then, the driver will have already started losing control on the slippery ice. We need a system to disseminate black ice warnings to (potentially) approaching vehicles so they can react much earlier, rather than when they get very close to the black ice and have no possibility of reacting to it. Advance notification can help drivers re-route their travel to avoid the situation, stop or even cancel the trip. The black ice notification is symptomatic of other vehicular safety notification applications that require timely delivery of relevant information to just the right recipients – one whose needs are ideally met by an information centric network that truly exploits the available information from sensors on vehicles to enhance safety of all the vehicles on the roadways.

Let us examine the application in slightly greater detail. A black ice notification needs to reach vehicles well before the time they hit the black ice, both temporally and spatially, e.g., 10 minutes and/or a few kilometers prior to reaching the road segment covered with ice. The warning needs to be propagated to those vehicles approaching it. Thus, a locally ad hoc approach, where vehicles can only talk to vehicles and infrastructure nodes in their range (e.g., using their wireless link, maybe over a 200-300 meter range that IEEE 802.11.p has), is insufficient since it does not provide the temporal and spatial "cushion" (you have to be close to "hear" them, and by that time it is too late to react). Ad hoc multi-hop forwarding, which makes it possible to extend the range, is undesirable since it is based on flooding, which not only congests the network (e.g., imagine having 100 other safety notification applications in addition to black ice), it wastes time and resources on the receiving vehicles. They have to process many messages they may not have asked for, or are not interested in it, and may not even need it. "Selectiveness" of messages delivered to the right receivers is a key requirement.

One possible solution might be to use the traditional client/server-based model over the Internet, e.g., querying Google if there is a black ice situation on the route. While this solves the lack of selectiveness with flooding in ad hoc networks, such server-based solutions pose two issues: *1*) In an IP-based network, every vehicle interested in that information will have to query the server, which may reside at a distance, through an end-to-end connection. Thus, in reality, we might have hundreds of cars asking for the exactly same piece of content, each establishing a separate connection with the remote server, resulting in excessive traffic. This will result in delays and will waste server and network resources. *2*) Every vehicle has to query for the black ice condition on every road it will travel on, for every time instant it "thinks" that information is needed, with most if not all of the queries yielding no new, useful information. Further, picking the right polling frequency/period is likely to be an issue.

Information-Centric Networks (ICN [1], [2]) on the other hand, and especially Pub/Sub over ICN [3], overcome these problems. Since ICN enables in-network caching and aggregation and a means for accessing content regardless of its loca-

tion, there is no need for a separate end-to-end communication context with each vehicle. Coupled with a Pub/Sub service, subscribing vehicles will receive such content only when they need it and at the time they need it, saving considerable network and vehicle resources.

In this paper, we propose V-ICE, a decentralized, information-centric, Pub/Sub based message propagation architecture for vehicular safety. In V-ICE, safety messages have names that reflect their time and position. Vehicles publish the safety message to the nearby roadside unit (RSU) when they encounter black ice. They also subscribe to a particular subset of the whole namespace (based on route plan and time they expect to travel over a road segment). When it becomes likely that they will encounter black ice on their path, a nearby RSU relays relevant information to interested vehicles, based on their subscription. With a hierarchical namespace, subscribers can pick the right granularity of information to receive, based on the hierarchical level in the name space. Thus, they receive all the messages generated under that level, in a timely manner. We evaluate the performance of V-ICE with a trace-driven simulation on a realistic, city-level roadway system during a rush-hour interval of 4 hours. Our results demonstrate the improved efficiency and timeliness of V-ICE compared to a polling-based approach or with flooding.

II. Background & Related Work

A. Vehicular Networks

Vehicular communication is an important part of Intelligent transportation systems (ITS) that mainly aim at providing the means for wireless communication among vehicles, treating them as mobile nodes, to exchange warning messages in order to prevent and alleviate the impact of accidents and hazardous situations [4]. There are three main patterns of communication in a vehicular environment: V2V (vehicle-to-vehicle), V2I (vehicle-to-infrastructure) and I2V [5], [6]. Each vehicle can function both as an end user and a router, through their On-Board Units (OBU) as networking devices and Application Units (AU) that host particular service applications. Vehicles are also equipped with Global Positioning System (GPS) and special sensors so they can detect and report road conditions and hazards. RSUs provide services to those vehicles that are in their range. Both RSUs and vehicles collaborate in warning propagation throughout a large area so vehicles can send and receive safety messages and react faster and avoid accidents. Vehicular Ad Hoc Networks (VANET) use Dedicated Short Range Communication (DSRC) for message exchange between nodes [5], [7]. Vehicular messages can be sent over vehicular communication protocols or can use cellular/wireless communications directly to access the Internet without having to communicate through RSUs and network infrastructure.

A number of essential vehicle safety applications are described in [5], [8], [9]. These applications can be broadly classified as those related to intersection collision avoidance, public safety, sign extension, vehicle diagnostics/maintenance and information from other vehicles. These application categories have different allowable temporal/spatial cushions (e.g. timeliness criticality degree. required distance travel of messages, etc.), communication pattern (i.e. V2V, V2I, I2V, I2I or hybrid) and domain scope (i.e. in-vehicle, between vehicles, infrastructure or domain). The network needs to deal with large scale (numbers of vehicles and RSUs) and a significant amount of mobility (albeit with some predictability, because vehicles are constrained to be on the road) [9]. There is a strict real-time requirement for safety applications, as every second matters when it comes to dealing with avoiding an accident. A message arriving or getting processed even a little late will be of no use. Thus, latency is of paramount importance.

Safety messages may be sent either periodically or be event-driven. Most solutions for delivery are based on broadcast using flooding or multicast/unicast based on (IP) address or position [10], [11]. Both IP-based or geographical delivery/routing methods couple the notion of content and location. More recently, an information-centric approach [12] has focused on how to name content in such environment, e.g., how to divide the environment into named segments.

B. Information-Centric Networks

Information Centric Networks focus on separating identity from location and are particularly suitable for timely delivery of relevant information to recipients. Two architectures that we consider to be suitable in this context are Named Data Networking (NDN) [1], [13] and MobilityFirst (MF) [2]. While NDN uses hierarchical names, MF uses flat Globally Unique Identifiers (GUIDs) to identify objects (content, device, user, etc.). While NDN is focused on a query-response service interface, extensions to the model with a Content-Oriented Publish/Subscribe System (COPSS) [3]. We exploit COPSS for efficiently disseminating black ice incident encounters by vehicles. MF uses a Global Name Resolution Service (GNRS) to map GUIDs to Network Addresses (NAs), and uses late-binding to help support mobility of the recipient of a message and combines hop-by-hop store and forward to deliver information even when the receiving node is disconnected for a period of time and then reconnects at a different attachment point with a new NA. We exploit these capabilities in our design for disseminating vehicular safety information.

There have been efforts to adopt NDN for vehicular applications, particularly in a V2V context. The focus of the previous work (e.g., [14]) has been to efficiently deliver NDN-Interests to the appropriate publisher to fetch content. A primary focus has been to identify publishers by their geographical location and request content from them. While there are some similarities (our approach also recognizes the need to incorporate geographical information into the name), there are significant differences. We believe it is important to name the event, rather than the publisher, to allow for obtaining event information related to a geographical location. Moreover, we are critically dependent on time as a component of the name space. Finally, unlike the previous work, we are concerned with timely delivery of information, instead of just delivering a request for a content based on the shortest V2V path along a path. As we have shown, the ability to have a pub/sub service interface is key to the timely delivery of safety information.

III. Architecture & Design

The architecture of V-ICE depends on V2I/I2V communication for disseminating safety information. When a vehicle's sensors detect a black ice event (wheels slip, brakes and

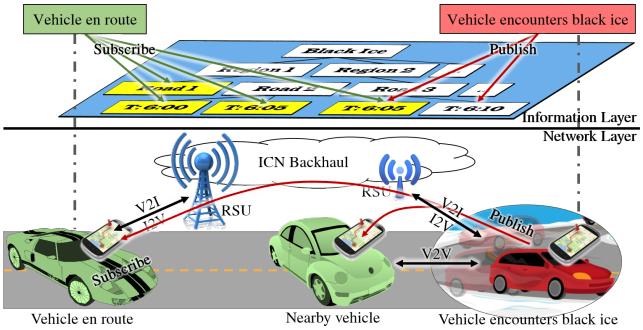


Fig. 1: Overall architecture of the proposed system.

steering are ineffective), they transmit a notification to the nearest RSU. Associated with it is a name that provides the geolocation and time, and a prediction for how long it is likely to last. RSUs are connected through a backhaul link to a (potentially multiprovider) backbone network. Warnings generated from a publishing vehicle are then delivered to interested subscriber vehicles over the infrastructure network. V-ICE's architecture can be enhanced by adding V2V communication in areas where the availability of RSUs is very limited. Fig. 1 shows overall architecture of V-ICE.

A key component of V-ICE's architecture is the name space that enables identification of the geographical location of the black ice events, along with a time interval for the event. The name space is organized as a hierarchy, to allow for aggregation of events that occur in a wider region. Subscriptions from vehicles (possibly an application such as a route planner with a GPS) will be for a number of names, corresponding to the road segments the vehicle will travel, along with the approximate time period it expects to be on that segment. The application can update or generate new subscriptions as the vehicle travels, based on the route, current position and speed. With the rich naming framework that supports hierarchies, subscriptions can use an aggregated name. Thus, a black ice event on any road segment within a region will result in that notification being delivered to a vehicle subscribed to events in the region. This will assist in re-routing the vehicle without selecting any road segments with black ice in that region.

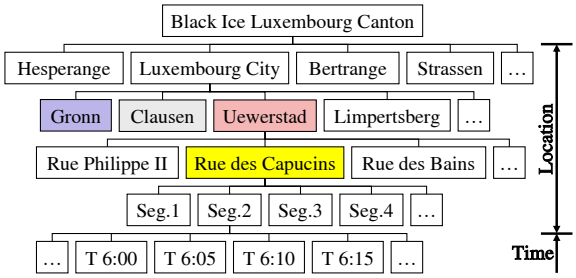
The architecture explicitly recognizes the need to support the delivery of safety warnings to vehicles that may have been disconnected from the network when the notifications was generated. We use the capability of an in-network store-and-forward reliable transfer introduced by the MobilityFirst [2] architecture. We address this in greater detail in Section III-B.

To further reduce the amount of network traffic generated, RSUs and routers in the backbone network filter messages based on names, so that duplicate events generated by vehicles publishing a black ice event (e.g., they all encounter the same event within a short time interval) don't deliver multiple messages to subscribing receivers. This is another benefit compared to an IP-oriented network, saving both network and client processing resources. More in Section III-C.

A. Naming Schema

The name space design assists drivers to precisely determine where black ice occurs, at what time and for how long. It also supports aggregated subscriptions/queries in order to reduce the subscription request load. The naming schema for V-ICE

(a) Map and partitioning in Luxembourg city



(b) Namespace example for Luxembourg canton

Fig. 2: Namespace design of V-ICE.

contains `/BlackIce/location/T:<time>`. An example of the namespace is shown in Fig. 2 (Fig. 2a shows the roads in Luxembourg and the partitioning into regions. Fig 2b shows the corresponding namespace).

The **location** component identifies the exact road segment. In a publication, it specifies where the black ice event is detected. E.g., in Fig. 2a, to travel on Seg.2 of Rue des Capucians, or to report a black ice event on that segment, the drivers can use the name `/.../Rue des Capucins/Seg.2`. To enable vehicles subscribe to the surrounding area of a road segment, i.e. nearby roads, for a certain period of time, we can use the higher level prefix E.g., `/.../Uewerstad`.

However, it is not sufficient to just specify the road segment. A subscription without a time component could result in receiving a lot of useless messages, e.g., black ice notifications that may be irrelevant by the time the vehicle reaches that road segment (e.g., melting). When planning the route it needs to specify the vehicle will traverse a road segment sometime later. Also, it is not appropriate to have a vehicle subscribe only to current events at short distances ahead on its path. It would likely seek to make decisions in advance. E.g., if the event occurs on a highway, the vehicle may want to re-route before entering the highway. Thus, it is important to include a time component in the naming scheme and use it in publications and subscriptions.

The **time** component in the name divides time into slots (of say, 5 minutes). To publish or subscribe, cars will use the time at the beginning of the slot as the time part of the name. E.g., to report a black ice at 6:03 which might last till 6:12, the car can publish with names `/.../T:6:00`, `/.../T:6:05`, and `/.../T:6:10` to indicate the period (the payload of could identify the time more precisely as 6:03–6:12 being the estimated time). Vehicles traveling through the area between 6:00 and 6:15 subscribe to those names and will receive the warning for that particular black ice event.

B. Reliability for Pub/Sub:

While it is attractive to assume that vehicles will be continuously connected to the infrastructure, there will inevitably

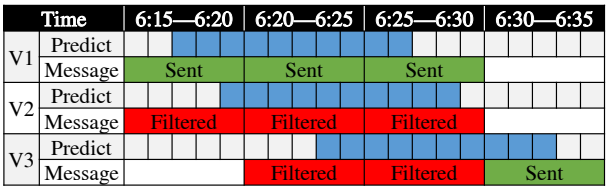


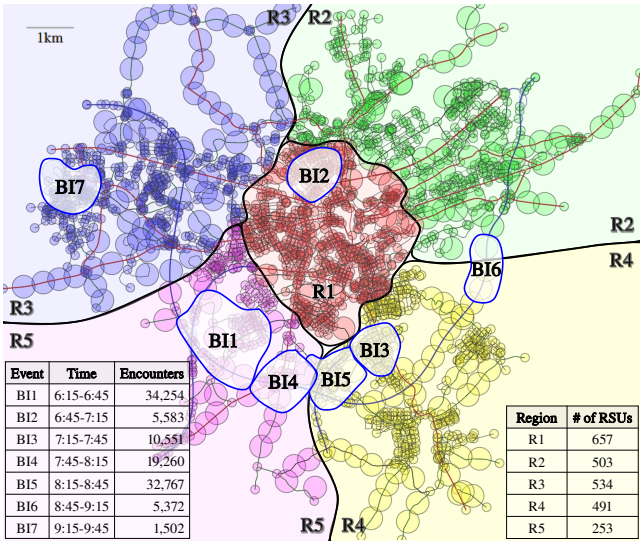
Fig. 3: Example of message filtering.

be short time intervals where a vehicle may be too far from an RSU (e.g., rural areas, highways with RSUs farther apart) and thus miss critical warnings. There are many solutions to ensure the reliable delivery of published messages with the pub/sub. One option is to use a network resident broker to store messages sent to recipients that subsequently reconnect (as in COPSS). Another alternative is for network routers to perform store-and-forward (as in MobilityFirst). This also allows for delivery of past notifications to a new vehicle coming into the system (e.g., start of a trip, route-planning) in addition to when a vehicle disconnects from one RSU and connects to another RSU after a few seconds. During the disconnected period, any publications generated will be buffered and delivered when the vehicle connects. Network routers can also perform late-binding – resolving names of objects to network addresses as packets are forwarded hop-by-hop – enabling mobility and use of a new network address for the recipient. We utilize the second solution as it does not require extra brokers and the messages can be stored at any hop.

C. Message Filtering

When a number of vehicles encounter the same black ice event, the same warning would be published by each one. This results in more network traffic, and more importantly, received by subscribed vehicles. It can overload receiving vehicles, and possibly distract drivers. It is desirable that these warnings be filtered. This filtering may be done at the subscriber, in the network, or at the publisher. Filtering at the subscriber side may be inappropriate as it consumes network traffic and resources on the subscriber vehicle. Having the publishing vehicle store previous warnings from others and suppress its own message may be difficult, due to storage and computation limits. It is also likely that the vehicle might have missed previous warnings because it was itself disconnected for a period of time. Filtering in the network (especially at RSUs) is more likely to be a reasonable option, utilizing the RSU's storage, compute power and continuous reliable connectivity. Determining duplicate notifications is also feasible based on the name in the message, without interpreting the payload.

Fig. 3 shows the message filtering design. An event on a road segment occurs starting at 6:15. The first vehicle V1 encounters it at 6:17. It predicts that the event would last at least 10 minutes, until 6:27. Since time is broken up into 5 minute slots, it sends out 3 warnings covering the interval 6:15 to 6:30. A second vehicle, V2, encountering the black ice soon after, at 6:19 at the same road segment would generate the same messages, but these would now be filtered by the RSU (having the same name). When vehicle V3 encounters the black ice at 6:23, it predicts that the event would last 10 minutes and generates warnings for the interval 6:20 to 6:35 (5 minute slots for the predicted period). The RSU would only



Event	Time	Encounters
BI1	6:15-6:45	34,254
BI2	6:45-7:15	5,583
BI3	7:15-7:45	10,551
BI4	7:45-8:15	19,260
BI5	8:15-8:45	32,767
BI6	8:45-9:15	5,372
BI7	9:15-9:45	1,502

Region	# of RSUs
R1	657
R2	503
R3	534
R4	491
R5	253

Fig. 4: Map partition & black ice events in simulation (based on LuST).

propagate the warning for the last 5 minute slot, the only non-duplicate announcement.

We have to extend the single-RSU suppression mechanism. For example if there are two RSUs R_1 and R_2 on the same road segment and 2 vehicles encounter black ice when communicating with R_1 and R_2, they might be experiencing the same black ice event. The RSUs should filter one of the messages. However, while this may be appropriate for a black ice event, there may be other safety applications where they should be seen as different events and not be suppressed.

Therefore, applications naming events appropriately is crucial, and policy would help RSUs determine if a message should be suppressed or not (similar to in-network aggregation). This is one of the benefits of using ICN – the network can provide more assistance to applications, and applications can give indications to the network using the right names.

D. Vehicle Route planning & Rerouting

Prior to departing, vehicles plan their route taking into account road conditions and congestion. Vehicles can also reroute on receiving warnings en-route. Routers that perform store-and-forward also maintain a cache of publications for an interval (based on the time specified in the publication) so that vehicles can query for the current status of road segments and receive responses from a in-network cache.Additionally, it is desirable for the vehicle to query for a larger area (e.g., region) to help in route planning. As with route planning, a vehicle may wish to perform a fast reroute (based on its local cache of information) upon receiving a black ice warning. By subscribing to a larger area, the vehicle is notified of events in the surrounding region of his planned route, and can use this information to re-route more efficiently.

IV. Evaluation

A. Simulation Setup

Simulation of Urban MObility (SUMO) [15] is an open-source traffic simulator for modeling different vehicle mobility patterns and based on synthetic or real road topologies. It can include traffic policies and rules for lane changes, traffic lights, etc. Coupled with a network simulator, SUMO can be a helpful tool for evaluating vehicular communication. Specifically, We

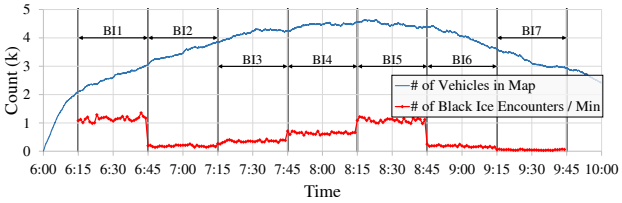


Fig. 5: Statistics of simulation setup.

use the Luxembourg SUMO Traffic (LuST) [16] scenario. We chose this because it is on a map of a city of a reasonable size, supports different traffic demands and levels of congestion, different road types (e.g., residential, arterial and highways), and different types of vehicles (passenger, public transport). Most importantly, we believe it is reasonably realistic for evaluating critical vehicular safety message dissemination applications, such as the black ice warning propagation we consider here. [16] shows that the LuST scenario behaves similar to a typical traffic pattern in the same area generated by Google Maps.

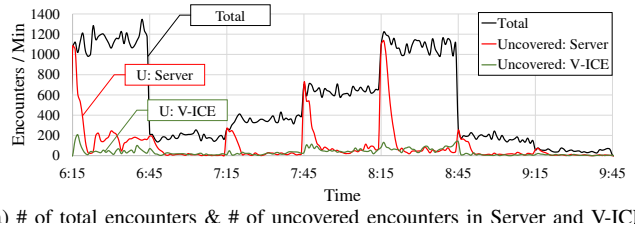
LuST has a raw map describing each road (and lane) in Luxembourg. In order to make this map a base for our vehicular infrastructure, we place RSUs, each with 100 meter range for wireless coverage, at 2,247 junctions. To improve total coverage throughout the city, especially for longer-distance highways where junctions of either endpoints are too far from moving vehicles, we add an additional 191 "edge" RSUs at 500 meter intervals on those highways. These special edge RSUs have a 250 meter range. The coverage was chosen based on the 802.11p standard for V2I communication [17], whose range, using the 5.9 GHz band with a signal strength threshold of 30 dBm, is 300 meters. LuST models a large number of distinct vehicles and their routes with start and 'leave' times. We export a 4-hour trace – 6am-10am, rush hour period.

We break the geographical area into regions and Fig. 4 shows the position of all the RSUs with the colors showing the region of each RSU. As shown in Fig. 4, there are still a few points on the map not covered by any RSU, which is reasonable as 100% coverage is likely not feasible in the real world. Vehicles not near any RSU will be disconnected. When they reach a point where they can again attach to the network via an RSU nearby, they re-connect. Each of the 5 regions has a representative top-level (border) router connected to five in-region mid-level routers. Each mid-level router is connected to approximately an equal (20%) number of RSUs in that region with wired links. The five top-level border routers are connected in a mesh. The network is a three-level hierarchy. The RSUs in each region are disjoint sets, i.e., each RSU is connected to one mid-level router (in the future we'll consider failure resiliency with multihoming). Fig. 4 shows the partitions (R1–R5) and # of RSUs in each. Fig. 5 shows the total number of vehicles over time (79,953 in 4-hour period).

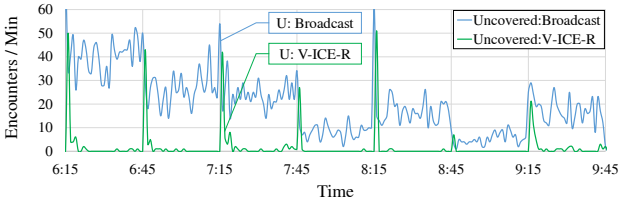
We first created 7 black ice incidents (BI1-BI7 in Fig. 4), starting at 6:15, each one lasting 30 minutes. 2 incidents are on the highway around Luxembourg (>30k vehicles hit black ice in half hour). Two more medium incidents occur on off-ramps and the city center (10k-20k vehicles), two small (∼5k) and one rural incident (∼1.5k). The figure also lists the number of vehicle encounters with black ice for each event. Fig. 5 also shows the number black-ice encounters (in thousands) per

TABLE I: Results: 30min and 6min scenarios.

	Solution	Encounters		Messages	
		Σ	Uncovered	Vehicle	Server/RP
30 min	V-ICE	109,289	7,160 (6.55%)	136,780	1,603
	V-ICE-R		349 (0.32%)	145,758	1,603
	Server		21,088 (19.30%)	222,241	331,768
	Broadcast		4,289 (3.92%)	5,990,676	—
6 min	V-ICE	110,303	11,293 (10.24%)	305,428	2,526
	V-ICE-R		1,603 (1.45%)	328,631	2,526
	Server		72,844 (66.04%)	217,699	332,782
	Server-H		4,299 (3.90%)	11,071,519	11,237,514
	Broadcast		5,061 (4.59%)	5,547,899	—



(a) # of total encounters & # of uncovered encounters in Server and V-ICE



(b) # of uncovered encounters in broadcast and V-ICE-R

Fig. 6: # of encounters/min in 30min simulation.

minute over the 4-hour trace. In a second, more challenging scenario, we create more frequent black ice events (each event lasts for 6 minutes; all 7 events happen in a round-robin fashion, and occur 5 times). We name the two scenarios as "30min" and "6min", respectively.

The black ice event is reported by vehicles that encounter it using a name `/region/RSU/time:`. Time is broken up into 5 minute slots, and each report has a time stamp associated with a 5 minute time slot. A vehicle encountering black ice makes a prediction that the situation will last 10 minutes and therefore sends out at most 3 warning reports as shown in Fig. 3. E.g., `/region/RSU1/6:00:00` (means there is a black ice in the region of RSU1 between 6:00:00 and 6:04:59.

B. Simulation Results

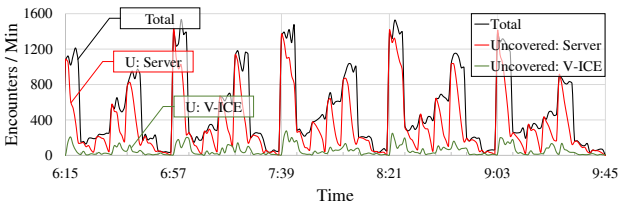
We compare V-ICE with a number of different alternatives for vehicular safety information dissemination, including using the server-based solution with clients polling, and a local flooding approach (as in V2V, VANET communication). We also compare the reliable pub/sub version of V-ICE, which we call V-ICE-R. For the server-based solution, the server collects warning reports over a 2 min indexing period and responds to queries from vehicles. Each vehicle generates a query every 5 minutes for road conditions on its route. In total, there will be a 7 min. latency for delivery of the warning notification to vehicles. However, to further examine the ability of the server solution to respond in a timely manner, we also look at the servers indexing the reports once ever 5 seconds, and the vehicles polling the server every 5 sec. We consider this the "high frequency" server solution, denoted as "Server-H". This trades-off improved latency against increased network load.

The metrics used in the evaluation are: *1*) The number of vehicles that hit the black ice without a notification in advance ('uncovered' encounters). This is the key metric of

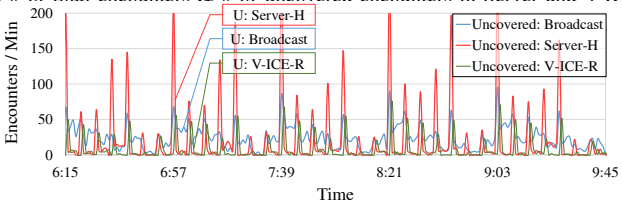
concern. *2*) the amount of network traffic generated, both for warning reports and for vehicles polling the server. With the "Broadcast" solution, we count each message transmitted by a vehicle encountering a black ice event. Ad-hoc forwarding by other vehicles or RSUs are not counted, to present the V2V solution in the most favorable light. *3*) the maximum number of messages received by a vehicle per minute. Fig. 6a shows the number of vehicles encountering a black ice event (in the 30min scenario) when there is no notification ('Total', with a peak of approx. 1200) vs. the server-based solution (Server) and V-ICE. Server brings down the number of 'uncovered' events (vehicles not receiving a notification in time) over time, but at the initial occurrence of the black ice event, the number of 'uncovered' vehicle events is almost the same as the raw Total. On the other hand, V-ICE reduces the peak down at least by a factor of 5 (to about 200 vehicle events), and quickly disseminates information to vehicles so that the number of uncovered vehicle events is very small. Fig. 6b also shows the improvement with the 'Broadcast' solution which brings down the number of 'uncovered' vehicle events further (to around 30), with a peak of 60. However with V-ICE-R, the reliable pub/sub alternative, the number of uncovered vehicle events is near zero, except at the very onset of the black ice event (with the first vehicle experiencing it). This shows the dramatic improvement with V-ICE-R. Table 1 shows the summary results (Total number of events is approximately 109K) of various options, with V-ICE-R bringing down the uncovered vehicle events down to 0.32% of the total. In terms of number of messages, V-ICE substantially reduces it, to just a little over the number of raw events. On the other hand, the server based approach generates almost 5 times more messages. Finally, the broadcast approach generates dramatically higher # messages.

With the higher frequency of black ice events – the 6min scenario – the raw total number of vehicle events goes up slightly, and the peak is higher at 1600. Because of the slower reaction of the Server approach, the peak number of uncovered vehicle events is about the same, demonstrating the ineffectiveness of the Server approach. However, V-ICE's peak uncovered events per minute is about the same as the 30min case, with a peak of about 200 per minute. We need the high frequency Server-H (poll every 5 seconds, process every 5 seconds at the server) scheme to approach the performance of V-ICE. But, V-ICE-R is much better. The uncovered vehicle events is no worse, despite the higher frequency of events. The total number of uncovered events over the 4 hour period (shown in Table 1) with V-ICE-R is approximately 1.45%, dominated primarily by the first vehicle encountering the black ice. Broadcast is not as good as V-ICE-R, having a higher peak of uncovered vehicle events, at the expense of tremendous communication overhead. With broadcast, the number of notification messages to be processed per vehicle per minute is of the order of 660 for the 6min scenario (636 for 30min scenario). The server-H solution requires the server to process approximately 12 million messages over a 4 hour period, yielding a processing time budget of about 1 millisecond per message at the server.

Overall, V-ICE-R (publish/subscribe with reliable delivery of publications to receivers after they come back online) performs far superior to the other options, in terms of timeliness (fewer uncovered vehicle events) and message overhead.



(a) # of total encounters & # of uncovered encounters in Server and V-ICE



(b) # of uncovered encounters in broadcast, V-ICE-R, and Server-H

Fig. 7: # of encounters/min in 6min simulation.

V. Conclusion

This paper proposes V-ICE, an information-centric vehicular safety communication architecture, studying in particular the case of 'black ice' information dissemination. Using publish/subscribe and a hierarchical namespace, V-ICE enables vehicles to receive all relevant safety messages in a timely manner just when it is needed. Our trace-driven simulations on a SUMO-generated rush-hour traffic in the city of Luxembourg show the improved efficiency and timeliness of our approach compared to a server-based polling oriented solution as well as a broadcast-based approach. Publish/Subscribe needs to have the appropriate support to ensure that vehicles that are disconnected still receive safety notifications when they come back online. With that, V-ICE-R dramatically reduces the number of vehicles that will suffer the serious consequence of an accident because it did not get notified of a black ice event in its path.

References

[1] V. Jacobson *et al.*, "Networking Named Content," in *CoNEXT*, 2009.
[2] D. Raychaudhuri *et al.*, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 3, pp. 2–13, 2012.
[3] J. Chen *et al.*, "COPSS: An Efficient Content Oriented Pub/Sub System," in *ANCS*, 2011.
[4] "Vehicular Safety Communications Project Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled DSRC," N.H.T.S. Administration, Technical Report DOT HS 809 859, March 2005.
[5] M. C. Weigle and S. Olariu, *A routing scheme for content-based networking*. Chapman and Hall/CRC, 2009.
[6] "Car2Car Communication Consortium," https://www.car-2-car.org/index.php?id=5.
[7] D. Jiang *et al.*, "Design of 5.9 GHz DSRC-based Vehicular Safety Communication," *IEEE Wireless Communications*, pp. 36–43, 2006.
[8] "Vehicle Safety Communications - Applications (VSC-A): Final Report," N.H.T.S. Administration, Technical Report DOT HS 811 492A, September 2011.
[9] S. Al-Sultan *et al.*, "A Comprehensive Survey on Vehicular Ad Hoc Network," *Journal of Network and Computer Applications*, p. 380–392, 2014.
[10] R. Baldessari *et al.*, "NEMO meets VANET: A Deployability Analysis of Network Mobility in Vehicular Communication," in *ITST*, 2007.
[11] T. Imielinski and J. Navas, "GPS-Based Addressing and Routing," IETF RFC 2009, November 1996.
[12] S. Kumar *et al.*, "CarSpeak: A Content-centric Network for Autonomous Driving," ser. SIGCOMM, 2012.
[13] L. Zhang *et al.*, "Named Data Networking (NDN) Project," PARC, Tech. Report NDN-0001, 2010.
[14] G. Grassi *et al.*, "Navigo: Interest Forwarding by Geolocations in Vehicular Named Data Networking," in *WoWMoM*, 2015.
[15] D. Krajzewicz *et al.*, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, pp. 128–138, 2012.
[16] L. Codeca *et al.*, "Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research," in *VNC*, 2015.
[17] J. Gozálvez *et al.*, "IEEE 802.11p Vehicle to Infrastructure Communications in Urban Environments," *IEEE Communications Magazine*, 2012.