WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2010; **10**:270–291
Published online 24 March 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.761

On the characteristics of routing paths and the performance of routing protocols in vehicle-formed mobile ad hoc networks on highways

S. Y. Wang*,†, C. L. Chou and C. C. Lin

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

Summary

Recently, intelligent transportation systems (ITS) is becoming an important research topic. One goal of ITS is to exchange information among vehicles in a timely and efficient manner. In the ITS research community, inter-vehicle communications (IVC) is considered a way that may be able to achieve this goal. An information network built on the top of vehicles using IVC can be viewed as a type of mobile *ad hoc* networks (MANETs). In the past, several unicast routing protocols for MANET have been proposed. However, most of them are designed for general MANETs rather than for IVC networks.

In this paper, we first used more realistic vehicle mobility traces generated by a microscopic traffic simulator (VISSIM) to understand the characteristics of routing paths in an IVC network. Based on the insights gained from the derived path characteristics, we designed and implemented an intelligent flooding-based routing protocol for small-scale IVC networks. *Via* several field trials conducted on highways, we compared the performance of *ad hoc* on-demand distance vector (AODV) and our protocol. Our experimental results show that (1) our protocol outperforms AODV greatly in IVC networks and (2) our protocol can provide text, image, audio, and video services for small-scale IVC networks (e.g., a platoon) quite well. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: inter-vehicle communication; intelligent transportation systems; mobile *ad hoc* networks

1. Introduction

In recent years, intelligent transportation systems (ITS) has become an important research topic. ITS aims to provide drivers with safer, more efficient, and more comfortable trips. For example, ITS wants to provide drivers with timely traffic congestion and road condition information so that drivers can avoid congested or dangerous areas. In addition, ITS wants to provide drivers with networking services so that they can exchange information, send/receive e-mails, browse web pages from the Internet, etc.

In the ITS research community, inter-vehicle communications (IVC) has attracted the interests of many automobile manufactures and researchers. In such a scheme, no infrastructure is required for communications between vehicles, and each vehicle is equipped with a wireless radio by which it can send and receive its own messages and forward messages for other vehicles. The vehicles on the roads dynamically form an

*Correspondence to: S. Y. Wang, Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

†E-mail: shieyuan@csie.nctu.edu.tw

ad hoc network at any time. Such an information network can be viewed as a type of mobile *ad hoc* networks (MANETs). In the rest of this paper, for brevity, we will simply call such a vehicle-formed MANET an IVC network.

In ITS, timely and efficient information distribution, acquisition, and exchange among vehicles is important. However, it is not easy to achieve these goals. First, an IVC network can easily get partitioned. This situation can easily happen when traffic density is low (e.g., at midnight), when the wireless transmission range is short, when few vehicles are equipped with wireless radios, etc. Second, a routing path established between a pair of vehicles in an IVC network can easily break because of frequent lane-changing activities on the roads. In a traditional unicast routing protocol design (such as *ad hoc* on-demand distance vector (AODV) [1]), a mobile node periodically issues control packets to detect and maintain neighboring nodes, find and set up routing paths, and repair broken paths, etc. Although this design works reasonably well for general MANETs, it may respond too slowly to fast-changing routing paths and encounter the following problems in an IVC network.

First, a routing protocol may not be able to detect the existence of a short-lived path and use it. In most routing protocols, for a mobile node to detect its neighboring nodes, each mobile node is required to broadcast its periodic HELLO packets to announce its presence. The time interval for these HELLO packets normally is set to a medium value to avoid excessive control packet bandwidth overhead. (e.g., the default HELLO interval specified in AODV RFC is 1 s) In addition, HELLO packets can be dropped quite easily due to wireless signal interference, fading, and vehicle blocking in IVC networks. These factors cause new neighboring nodes and paths to be detected slowly on IVC networks. If most paths in an IVC network have a lifetime shorter than the average time needed to detect a new path, a routing protocol will not be able to detect and use such paths.

Secondly, a routing protocol may not be able to detect the disappearance of a short-lived path quickly and thus will still keep sending data over it. As such, all of the data sent over this no longer existing path will be dropped. In most routing protocols, to overcome the unpredictability of wireless signal, usually a mobile node is designed to tolerate some successive losses of HELLO packets sent from a specific neighboring node before considering it disconnected. For example, the default value for this parameter is 2 in AODV RFC. With this design, however, the disappearance of a path cannot be detected immediately. Instead, it can only be detected after $hello\_interval * (allowed\_loss\_number + 1)$ s. Therefore, in AODV the time needed to detect a path loss is at least 3 s. If most paths in an IVC network have a lifetime shorter than the average time needed to detect a path loss, a routing protocol will spend most of its time on sending data over non-existing paths and waste a lot of network bandwidth.

Thirdly, a routing protocol will need to trigger its path repair design very frequently and this will cause a significant amount of control packet bandwidth overhead. Most routing protocols designed for MANET have their own path repair designs to repair a broken path. For example, in AODV, the routing protocol will first perform a local repair at the path breaking point trying to find a backup path from the breaking point to the destination node. If no such path can be found, an error packet will be sent to the source node to initiate a global path repair, which will try to find a backup path from the source to the destination nodes. Both of these designs use the scoped-flooding method to flood the path-search control packet over an area of the network. Since the flooding operation is very bandwidth-consuming, if generally an established path breaks very often in an IVC network, most of the IVC network's bandwidth will be consumed by these control packets.

Lastly, frequent path changes caused by frequent path repairs will harm the performance of the TCP transport-layer protocol greatly. For any routing protocol, there is a time gap between the time when a path breaks and the time when the path breakage is detected. In this period of time, however, several packets may have been dropped due to no route to the destination vehicle. In addition, when a backup path is found and used, several in-flight packets may get out-of-ordered due to traveling on different paths (i.e., the new backup path and the old one). For TCP, which is the most widely used protocol in the current Internet, because each lost packet or several out-of-order packets will cause the TCP sender's transmission rate to be reduced by a half (sometimes even causes the TCP sender to timeout for at least 1 s), constantly changing the routing path between the source and destination vehicles will harm TCP performance greatly.

From the above discussions, we see that the lifetime of routing paths in an IVC network affect the performance of the IVC network greatly. Thus, in this paper, we conducted simulations to understand the lifetime characteristics of routing paths in IVC networks.

The contributions of this paper are as follows. First, we used more realistic vehicle mobility traces to

DOI: 10.1002/wcm

derive the lifetime and path breakage characteristics of unicast paths in an IVC network. In the past, although the authors in Reference [2] also studied the lifetime of routing paths in *ad hoc* networks, their analysis is based on the assumption that the mobility of nodes can be described by some simple mathematical models (like the random waypoints model [3]). As such, their results are not applicable to real-world IVC networks. Second, we conducted field trials to evaluate the performance of AODV on the roads. In the literature, such performance data are rare. Third, we designed and implemented an intelligent flooding-based routing protocol for IVC networks. Results obtained from field trials show that our protocol outperforms AODV significantly in IVC networks. Lastly, our protocol can provide useful services such as e-mail, FTP, web, video conferencing, and video broadcasting for vehicle users in IVC networks. So far, such capabilities are still rare.

The rest of this paper is organized as follows. In Section 2, we survey routing protocols proposed for general MANET and investigate their drawbacks under high mobility conditions. In Section 3, we describe the simulation environment and settings. In Section 4, we explain the performance metrics used in this study. In Section 5, we present the simulation results. In Section 6, we study the issues of platoon communication in an IVC network. In Section 7, we discuss simulation results. In Section 8, we describe the architecture of our routing protocol. In Section 9, we elaborate on the protocol design. In Section 10, we present its implementation on the Linux operating system. In Section 11, we present the performance of our protocol in field trials. In Section 12, we present the important services that can be supported by our protocol. Finally, in Section 14, we conclude the paper.

2. Related Work

In the literature, several papers have discussed and studied the applications of MANET to IVC networks. We briefly describe them here. In Reference [4], the authors presented the framework and components of their 'Fleetnet' project, which aims to efficiently exchange information among vehicles. In Reference [5], the authors proposed a GPS-based message broadcasting method for IVC. In Reference [6], the authors proposed a GPS-based unicast routing scheme for cars using a scalable location service. In Reference [7], the authors showed that messages can be delivered more successfully, provided that messages can be stored temporarily at moving vehicles while waiting for opportunities to be forwarded further. In References [8,9], the authors studied how effectively a vehicle accident notification message can be distributed to vehicles inside a relevant zone. In Reference [10], the authors focused on how to establish a direct transmission link between two neighboring vehicles. In Reference [11], the authors proposed some changes to AODV routing protocols for IVC networks. In Reference [12], the authors compared the packet delivery ratio of a location-based routing protocol with that of a topology-based routing protocol on a simulated highway IVC network. In Reference [13], the authors proposed a position-based routing protocol for IVC networks in city environments. In Reference [14], the author studied the effectiveness of broadcasting information on an IVC network, and in Reference [15] the author studied the intermittence of routing paths in an IVC network.

Regarding the routing protocols proposed for general MANET, they can be classified into two categories: single-path routing and multi-path routing. Most routing protocols for MANET are single-path-based protocols, which can be further classified into two groups: 'proactive' and 'reactive'.

Proactive (also called 'table-driven') routing protocols, such as DSDV (destination-sequenced distance-vector routing protocol [16]), require nodes in an MANET to periodically exchange their local information (e.g., the states of neighbors, the number of neighbors, etc.) so that every node can have correct and consistent routing information. Propagating the freshest routing information throughout the whole network is used to update topology changes. When a table-driven routing protocol is used in a fast-changing IVC network, it may take a long time for the routing information to converge and may cause a lot of control packet bandwidth overhead.

Reactive (also called 'demand-driven') routing protocols, such as AODV [1] and DSR [17], attempt to discover a route only when needed. When a source node is going to send packets, a demand-driven protocol first floods a path-search control packet throughout the network to find available routes. After a route is found, the routing protocol needs to periodically detect whether the path has become broken so that a path-repair operation can be initiated quickly. Since routes in an IVC network are fragile, such a routing protocol may need to generate many control packets to quickly detect and repair broken paths. Otherwise, packets will be sent over non-existing paths and network bandwidth will be wasted.

Multi-path routing protocols, such as GeoTORA [18], are able to discover multiple routing paths for

DOI: 10.1002/wcm

data forwarding. Some multi-path routing protocols are the extensions derived from single-path routing protocols. For example, the protocols proposed in References [19–22] are extended from AODV, the protocols proposed in References [23] and [24] are the extensions of DSR, and the protocol proposed in Reference [25] is a variation that combines AODV and DSR to achieve multi-path routing. Although a multi-path routing protocol can potentially find a backup path to replace a broken path more quickly than a single-path routing protocol, maintaining multiple paths simultaneously may cause a lot of control packet bandwidth overhead in a fast-changing IVC network.

The flooding technique is commonly used in the research areas of broadcasting and multicasting [26–32]. These studies focus on how to achieve the largest coverage for a broadcast by the least number of duplicated packets. Most studies reduce the number of duplicated packets based on the information about neighboring nodes (e.g., their locations and their moving speeds) and assume that the information can be obtained by exchanging control packets. Our protocol can be classified into this kind of protocols. However, our protocol focuses on reliably flooding a packet from the source to the destination vehicles with the least number of duplicated packets. Maximizing the coverage with the least number of duplicated packets is not the goal of our protocol.

In the proceedings of ACM VANET2004 conference, several papers study data dissemination problem in IVC networks. In Reference [33], the authors studied the single-hop broadcast reception rates under two channel models and signal interference conditions. In Reference [34], the authors proposed a new broadcast protocol for IVC networks, which addresses broadcast storm, hidden node, and reliability problems in urban areas. In Reference [35], the authors studied the feasibility of using mobile vehicles as gateways for other vehicles in an IVC network. In Reference [36], the authors proposed a mobility-centric data dissemination algorithm for IVC network. This algorithm exploits vehicle mobility and combines the idea of opportunistic forwarding, trajectory-based forwarding, and geographical forwarding.

Recently, some location-aware routing protocols (e.g., [37]) have been proposed to use mobile nodes' location information to greedily route packets toward their destination nodes. Routing protocols of this type do not need to install states in mobile nodes to find a path, set up a path, and maintain a path as traditional routing protocols do (e.g., References [1] and [17]). Although they can better cope with high mobility problems, they have some disadvantages. First, for a mobile node to know the location of a remote mobile node, a location directory service is needed, which, however, is difficult to provide. Second, greedy routing may lead packets to 'dead branches' in the network, where these packets will need to be dropped due to no route to their destination nodes. Third, location-aware routing protocols still rely on exchanging periodic HELLO messages to detect neighbor appearance and disappearance. As such, they will also experience the problems discussed in Section 1 in highly mobile IVC networks. Due to these problems, the routing protocol studied in this paper is assumed to be a traditional routing protocol that does not use location information to route packets.

3. Simulation Settings

3.1. Traffic Simulator

The microscopic traffic simulator that we used to generate mobility traces of vehicles is VISSIM 3.60 [38], which is a commercial software developed by PTV Planung Transport Verkehr AG company, located in Germany. VISSIM uses the psycho-physical driver behavior models developed by Wiedemann [39,40] to model vehicles moving on the highways. This includes acceleration/deceleration, car-following, lane-changing, and other driver behaviors. Stochastic distributions of speed and spacing thresholds can be set for individual driver behavior. According to the user manual, the models have been calibrated through multiple field measurements at the Technical University of Karlsruhe, Germany. In addition, field measurements are periodically performed to make sure that updates of model parameters reflect recent driver behavior and vehicle improvements.

3.2. Highways System

The topology of the highway used in this study is depicted in Figure 1. It is a rectangular closed system with four circular corners and has three lanes in each direction. Its length and width are 8 and 5 km, respectively. There are no entrances and exits on this highway system. Vehicles are injected into this system in both directions at the top-left corner. The injection rate is 1000 vehicles per hour in each direction. After all vehicles have entered the system, they move freely in the highway system according to their respective desired speeds, vehicles characteristics, and driving behavior.

DOI: 10.1002/wcm

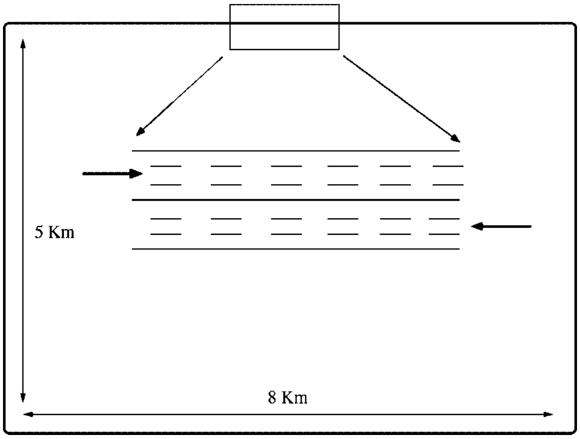

Fig. 1. The topology of the highway used in this study.

Although the chosen highway topology may not be very realistic compared with highways in the real world, we think that their difference is not important from the viewpoint of dedicated short range communication (DSRC) wireless transmission. It is true that a real-world highway may not look like a rectangle and instead may have several curves. However, for the safety of high-speed driving, the radii of these curves usually are very large (say, a few kilometers). This property makes these curves effectively equivalent to straight lines when the 100-m DSRC wireless transmission range is used.

Since vehicles are assigned different desired speeds and different thresholds for changing lanes for achieving their desired speeds, a vehicle may thus (1) move at its desired speed when there is no slower vehicle ahead of it, (2) follow the lead vehicle patiently, which may happen when the lead vehicle is slower but the difference between the lead vehicle's speed and its own desired speed is still tolerable, or (3) decide to change lanes to pass the lead vehicle if the speed difference is intolerable.

For each different simulation case, we took ten vehicle traces after all vehicles have entered the highway system and have been moving freely for one hour (3600 s). Each trace lasts for 300 s and the location information of all vehicles in every second of this period are logged into the trace. We took these traces consecutively. That is, the first trace is taken from the period ranging from 3601 to 3900 s the second trace is taken from the period ranging from 3901 to 4200 s, etc. For each of the 10 traces of a simulation case, we used the location information stored inside it to derive a performance result. Then, we computed and reported the average and standard deviation of these results. Doing so avoids picking up a trace that is not representative if we would have analyzed only a trace for a simulation case. Also, it can show whether a parameter value is sensitive to a particular trace.

Note that in this highway system, vehicles in different directions do not interact with each other. This is because in this topology a vehicle cannot leave the highway in one direction and then enter the highway in the opposite direction.

3.3. Vehicle Traffic

In this study, the total number of vehicles moving in the highway system is set to be 2000 and a half of them are moving in each direction. The average distance between a vehicle and the vehicle immediately following it on the same lane can be calculated as follows. It is (26 km/lane × 3 lanes/direction)/(1000 vehicles/direction) = 78 m, where 26 km is the perimeter of the simulated highway. This car following distance is typical of a highway in which many vehicles are using the highway but they are moving smoothly without congestion.

The desired speeds chosen for these vehicles determine the absolute speeds of these vehicles and the relative speeds among them. The distribution of these desired speeds is (20%: 100–110 km/h, 40%: 90–100 km/h, 20%: 80–90 km/h, 20%: 70–80 km/h), which means that 20% of the vehicles are moving at their desired speeds uniformly distributed between 100 and 110 km/h, 40% of the vehicles are moving at their desired speeds between 90 and 100 km/h, etc. We think that this distribution is typical of a highway in which various types of vehicles exist.

3.4. Wireless Radio

The transmission range of the wireless radios used in these vehicles is chosen to be 100 m. It is a reasonable setting for the DSRC standards proposed for ITS applications.

Since this paper focuses only on the connectivity among vehicles rather than the achievable data transfer throughput among them, this paper does not consider the bandwidth of wireless radios and the medium access control protocol used by them. Instead, we took a simplified approach to determine whether or not two vehicles can successfully exchange their messages. In our study, as long as two vehicles are within each other's wireless transmission range, their message exchanges will succeed. Otherwise, their message exchanges will fail. This scheme is similar to that used

DOI: 10.1002/wcm

in the ns-2 simulator [41], except that 250 m is used as the transmission range of IEEE 802.11 wireless LAN in ns-2.

4. Studied Performance Metrics

The studied performance metrics are described in this section. For each metric, we analyze and show its performances in four different cases, which are the combination of the studied path population and the policy used to find the shortest path between a pair of vehicles.

We classify all unicast paths in the highway system into two path populations and report their performances separately. We decide not to report the performances of the aggregation of these two populations because we found that some of their performances differ significantly and it is better not to mix them.

The first population consists of all of the paths whose source and destination vehicles move in the same direction in the highway system (SameDirection). The second population, on the other hand, consists of all of the paths whose source and destination vehicles move in the opposite directions in the highway system (DifferentDirection). It is clear that the aggregation of these two populations represents all of the paths in the highway system.

The tested policies include the normal policy and a new policy, which differ in how a shortest path between a pair of vehicles is found. In the normal policy, the distance weight of 1 is given to all wireless links before we computed the shortest path. In the new policy, 1 is given to all wireless links whose two vehicles move in the same direction while 100 is given to all wireless links whose two vehicles move in opposite directions before we computed the shortest path. The motivation behind testing the performance of the new policy is based on one observation that a wireless hop (i.e., wireless link) whose two vehicles move in the opposite directions generally breaks more easily than a wireless hop whose two vehicles move in the same direction. (For brevity, in the rest of the paper we will simply call the former links the 'ODL' and the latter links the 'SDL', where ODL stands for 'opposite direction link' and SDL stands for 'same direction link'.)

The reason for such a phenomenon is clear because when two vehicles move in the opposite directions and at high speeds, the period of time in which a wireless link can be set up between them is quite short. It is two times of the wireless transmission range divided by their relative moving speed. For example, suppose that both vehicles move at 100 km/h (27.7 m/s) and the wireless transmission range is 100 m, the lifetime of the wireless link between them is only 3.6 s $(2 \times (100/(2 \times 27.7)))$. On the other hand, if the two vehicles of a wireless hop move in the same direction, although they may have some small speed difference, the lifetime of the wireless hop is still relatively long. For example, suppose that the speed difference between the two vehicles of a wireless hop is 20 km/h (5.54 m/s), then the lifetime of the wireless hop will be 18 s, which is 500% improvement. For this reason, it may be wiser to use more reliable SDL links to construct a routing path rather than less reliable ODL links unless they are really needed.

Since each of these policies is applied to each of the path populations, we have four different cases in total. In the rest of the paper, they are named SameDirectionNormal, SameDirectionSDLPreferred, DifferentDirectionNomal, and DifferentDirectionSDLPreferred, respectively.

4.1. Lifetime Percentage Distribution

The first performance metric is the lifetime percentage distribution of all paths in a particular case. We define the lifetime of a repairable unicast path between two vehicles as the duration in which there exists one path between them. That is, during this period these two vehicles can find a path to exchange their messages, even though this path may need to be repaired during this period. We note that in our study the lifetime of a repairable path is determined solely by the vehicle mobility trace and is independent of the routing protocol used. In our study, if a path needs to be repaired, we use the global path repair design to repair it. (For brevity, in the following of this paper, we will simply use 'unicast path' or 'path' to represent 'repairable unicast path'.)

The unit of path lifetime is set to be second. Starting from the first second of a trace, for every pair of vehicles, we check whether a path can start in each second. We say that a path between two vehicles starts in *N*th s if there exists a path between them in *N*th s. Once a path is found (created) between two vehicles, in each subsequent second we then check whether it would break in this second. A path is considered broken if any wireless link (i.e., hop) of its path no longer exists. If the path does not break in this second, we repeat this connectivity test in the next second.

Suppose that a path is found to be broken in *M*th second, we will try to find the shortest backup path between the source and destination vehicles. If no such backup path can be found, the lifetime of this repairable unicast path is now determined and it is $(M + 1) - N$.

DOI: 10.1002/wcm

If such a path can be found, the old path is replaced with this new path and its connectivity is tested in each subsequent second as before.

Note that, according to the above definition, all paths that can be started in any second during the trace are accounted and processed separately. Also, although in this way each found path has a lifetime of at least 1 s, its exact lifetime actually may be less than 1 s.

The lifetime percentage distribution is useful. It gives us a sense of how long generally an established path can last. Clearly, we prefer to see long lifetime rather than short lifetime for these paths. Otherwise, many useful unicast-based applications such as e-mail, FTP, HTTP, and telnet are unlikely to be useful on an IVC network.

4.2. Lifetime Number Distribution

This performance metric is like the previous metric except that the distribution shows the number of the paths with a specific lifetime rather than their percentage. Although this metric is like the previous metric, it has its own value. This metric can show how easily or difficultly a path can be found (set up) in a case. For example, in a 300-s trace, if more paths can be found in a case than in another case, then it means that it is easier to find (set up) a path in the former case. In contrast, a lifetime percentage distribution lacks the number information and thus cannot show us how easily a path can be found in a case.

4.3. Path Repair Hop Count Difference *Versus* Lifetime

This metric is the relationship between the lifetime of paths and their corresponding path repair hop count difference. The path repair hop count difference of a path is defined as follows. Suppose that during its lifetime, a path experiences N successful path repairs and the hop counts of the initial path and these N reformed paths are $H_0, H_1, H_2, \ldots, H_N$, respectively. Then the path repair hop count difference of this path during its lifetime is defined to be $(ABS(H_1 - H_0) + ABS(H_2 - H_1) + \cdots + ABS(H_N - H_{N-1}))/N$, where $ABS()$ is the absolute value function. After the path repair hop count differences of all paths are calculated, for each specific lifetime, we then calculate the average of the path repair hop count differences of all paths whose lifetime is the same as that specified.

This information shows how many hops a path may vary between its successive path repairs. If the number is large, the packets of a greedy transfer are very likely to get out-of-ordered due to a large hop count difference between the old and the new paths. This will result in poor TCP performance.

4.4. Path Repair Time Interval *Versus* Lifetime

This metric is the relationship between the lifetime of paths and their corresponding path repair time interval. The path repair time interval of a path is defined as follows. Suppose that a path starts at T_0 and during its lifetime it experiences N successful path repairs at $T_1, T_2, T_3, \ldots, T_N$, respectively. Then the path repair time interval of this path during its lifetime is defined to be $((T_1 - T_0) + (T_2 - T_1) + \cdots + (T_N - T_{N-1}))/N$. After the path repair time intervals of all paths are calculated, for each specific lifetime, we then calculate the average of the path repair time intervals of all paths whose lifetime is the same as that specified.

This performance metric shows how frequently a routing protocol needs to use its path repair design to extend a path's lifetime. Clearly, we prefer to see a long path repair time interval; otherwise, constantly triggering the path repair design will incur much control packet bandwidth overhead and hurt TCP performance greatly.

5. Simulation Results

5.1. Lifetime Percentage Distribution

Figure 2 shows the cumulative distribution function (CDF) of the lifetime of the paths found in the four different cases. The CDF curves of these four different cases give us several insights. First, the lifetime of most

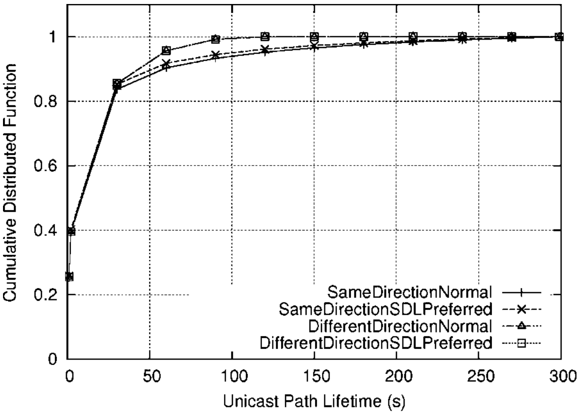

Fig. 2. The CDF of the lifetime of the paths on an IVC network.

DOI: 10.1002/wcm

repairable paths are very short. Over 70% of these paths have a lifetime less than 10 s. Second, these distributions are almost identical. This means that, if a path can be set up between two vehicles, no matter whether it is in the SameDirection or DifferentDirection path population, its expected lifetime is the same. Third, the proposed path-selection policy changes the percentage distribution only minimally. We have conducted more thorough analyses on the logged traces to find the reasons for the second and third findings. We found that to find a path for a pair of vehicles on the simulated highway, whether the path is in the SameDirection or in the DifferentDirection path population, most of the time the found path needs to contain at least one ODL. As such, no matter how we prefer SDLs over ODLs in the path-selection phase, eventually at least one ODL is still included in the found path. This property results in the phenomenon reported in the second and third findings.

In the figures, each case's curve represents the average (ave) of the results of its 10 different traces. In some figures in the rest of the paper, the standard deviation (std) is also shown as a vertical bar spanning the [y_low = ave − std, y_high = ave + std] range. With the standard deviation information, we can know whether the performance differences between two curves are significant and consistent.

5.2. Lifetime Number Distribution

Figure 3 shows the number distributions of the lifetime of the paths found in the four different cases. We see that these number distributions are also almost identical. The total number of paths in these four cases differ only by 2%. This means that when a vehicle wants to set up a path to another vehicle, it need not care about whether that vehicle is moving in the same direction with itself or not. The chance that a path can be set up between them will be about the same for these cases. We also conducted more thorough analyses on the logged traces to find the reasons for this finding. We found that the reason is the same as that explained above. That is, to find a path for a pair of vehicles on the simulated highway, whether the path is in the SameDirection or the DifferentDirection path population, most of the time the found path needs to contain at least one ODL. It is this property that makes the lifetime characteristics of paths in these four cases similar.

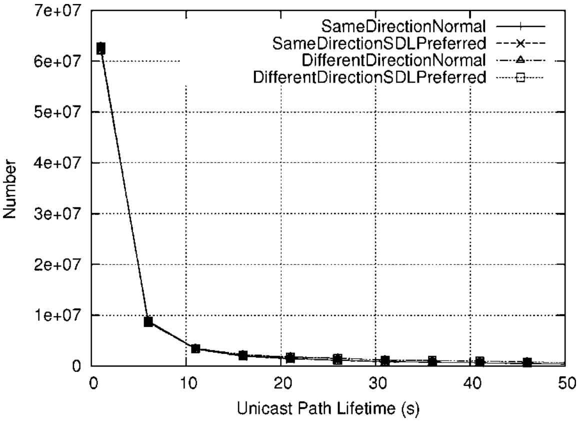

Fig. 3. The number distribution of the lifetime of the paths on an IVC network.

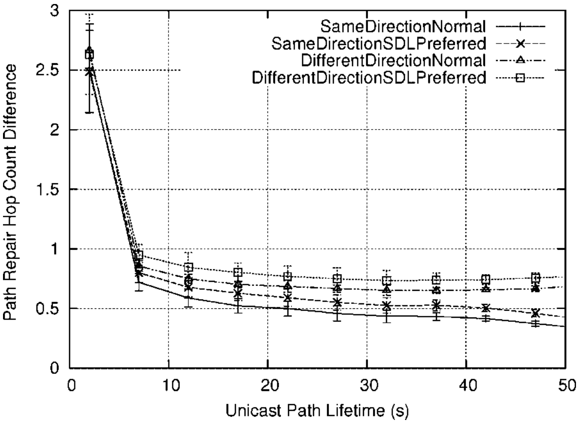

Fig. 4. The path repair hop count difference of paths *versus* their lifetime on an IVC network.

5.3. Path Repair Hop Count Difference *versus* Lifetime

Figure 4 shows the relationship between the path repair hop count difference of paths and their corresponding lifetime. We see that, beyond 4 s, the hop count difference between path repairs generally remains about the same and is less than 1 hop. This is a good news for TCP performance because packets are less likely to get out-of-ordered under this condition.

5.4. Path Repair Time Interval *Versus* Lifetime

Figure 5 shows the relationship between the path repair time interval of paths and their corresponding lifetime. First, from the absolute performance numbers of these curves, we see that the path repair time interval is very small and is only between 1 and 1.5 s. This is a bad news as the path repair design of a routing protocol will need to be triggered very frequently and thus cause

DOI: 10.1002/wcm

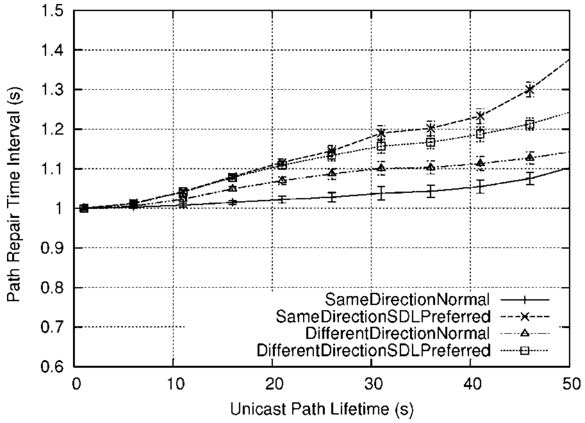

Fig. 5. The path repair time interval of paths *versus* their lifetime on an IVC network.

much control packet bandwidth overhead. Second, for the two cases using the new path-selection policy, we see that their path repair time intervals indeed are longer than those of the two cases using the normal policy. This result shows the effects of the new policy.

6. Platoon Communication

In this section, we focus on the communication of platoons moving on the highway system. According to the opinions of NISSAN Taiwan Inc. (which is cooperating with us), a platoon, formed by family members, friends, or colleagues, is the type of IVC network which is by far the most feasible application. This is because using IVC to exchange messages and conduct video conferencing among platoon vehicles does not need any communication cost. In contrast, if GSM/GPRS/3G cellular networks were used for these purposes, the communication cost would be very high. Thus, we investigate the characteristics of platoon communication using the simulation methodology and settings described before.

Because the VISSIM 3.60 traffic simulator does not support platoon movement simulation, we have to generate our own platoon movement traces and combine these traces with the 2000-vehicle VISSIM traces used before. By this approach, the moving behavior of platoon vehicles do not interact with those in the 2000-vehicle background traces. Although this approach is not perfect, we think that studying this combined trace still provides insights into platoon communication on an IVC network. In our own platoon movement traces, we let each vehicle, except the leading vehicle, follow its preceding vehicle closely. This is similar to the car-following behavior of a real-world platoon.

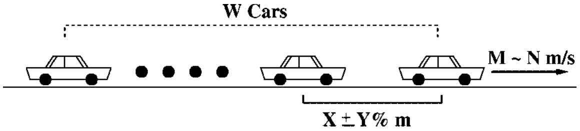

Fig. 6. The definition of a platoon.

Figure 6 shows our definition of a platoon that is used to generate our own platoon movement traces. In this figure, one sees that the formation of a platoon is based on some parameters, including the number of vehicles in a platoon, the (average) distance between two adjacent vehicles, and the moving speed of the leading vehicle. The leading speed and inter-vehicle distance are varied within specified ranges, respectively.

In the following simulations, the highway system adopted is the same as that shown in Figure 1. In addition to the original 2000 vehicles moving on the highway system, in each moving direction of the highway system, 10 platoons are inserted into the traffic. Each platoon has five member vehicles. In other words, there are in total 2100 ($2000 + 2 \times 10 \times 5$) vehicles moving on the highway system. The moving speed of the leading vehicle of a platoon ranges between 90 and 100 km/h. The (average) inter-vehicle distance is set to 100 m or varied as a system parameter, and its dynamic deviation in each second of the trace is set to $(+/-)$ 10% of the used average distance. The transmission range of the wireless radio is set to 100 m or varied as a system parameter.

In this platoon study, we study the characteristics of the routing path between the leading and the last vehicles of a platoon. The shortest routing path searching algorithm used in this platoon study does not favor platoon vehicles over non-platoon vehicles. For a platoon, vehicles not belonging to it may participate in the routing path between its leading and last vehicles. This can happen when two adjacent platoon vehicles are separated too far away and they need vehicles outside the platoon to relay their packets.

The inter-vehicle distance and the radio penetration rate system parameters are studied in our investigation. We study how the performance of platoon communication is affected by each of them below. The studied performance metrics are the average path lifetime and the average path repair time interval of the routing path between the leading and last vehicles of a platoon.

6.1. Inter-vehicle Distance

The first investigated system parameter is the (average) inter-vehicle distance. We varied the value of

DOI: 10.1002/wcm

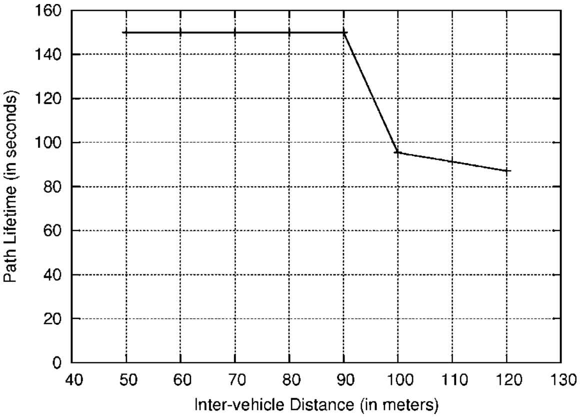

Fig. 7. The average path lifetime under different inter-vehicle distances.

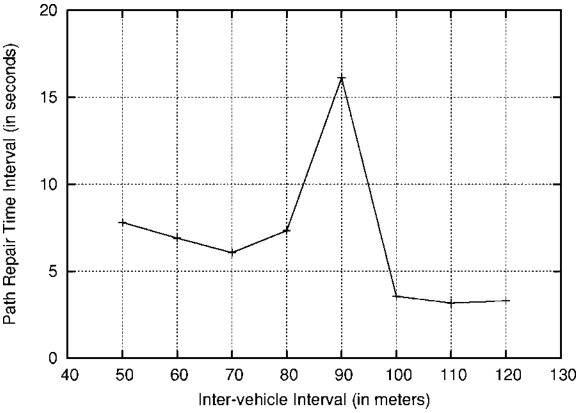

Fig. 8. The average path repair time interval under different inter-vehicle distances.

this parameter from 50 to 120 m in our investigation. Figure 7 shows the average path lifetime under different inter-vehicle distances.

The results show that when the wireless range is set to 100 m and the distance is set to a value less than or equal to 90 m, all of the built routing paths can last without breaking during the whole simulation time. (Note: an average lifetime of 150 s is the maximum possible average lifetime of a case in this study. This is because a path setup is attempted between the leading and last vehicles in each second of the 300-s trace in this study. Since the trace is only 300-s long and the path set up in the *N*th s of the trace can have only at most $(300 - N)$ s of lifetime. It is clear that the maximum possible average lifetime of a case is $(300 + 0)/2 = 150$ s.)

The reason for this curve is explained as follows. When the inter-vehicle distance is set to 90 m or less, the maximum dynamic inter-vehicle distance is $90 + 90 \times 10\% = 99$, which is less than the wireless range of 100 m. As such, the vehicles in the same platoon can provide a path between the leading and the last vehicle by themselves at any given time during the simulation. However, when the inter-vehicle distance is set to a value greater than or equal to the wireless range of 100 m, with the 10% dynamic deviation, sometimes the dynamic distance between two adjacent vehicles in the platoon may exceed the wireless range of 100 m. As such, the path between the leading and last vehicles may break, resulting a shorter average path lifetime. When the inter-vehicle distance increases to 120 m, as expected, because the chance of path breakage increases, the average path lifetime decreases further.

Figure 8 shows the average path repair time interval under different inter-vehicle distances. One sees that, except for the 80- and 90-m distances, the path repair time interval decreases as the inter-vehicle distance increases. This phenomenon can be explained below. When the inter-vehicle distance is set to 80 or 90 m, which is very close to the wireless range, the shortest path searching algorithm has a high chance to select platoon vehicles as the relay vehicles for the platoon. This is because doing so can make the resulting path the shortest path. Because the link connectivity between two adjacent platoon vehicles is more stable (because they move at about the same speed and in the same direction) than that between a platoon vehicle and a non-platoon vehicle, doing so also increases the stability of a found routing path and the resulting path repair time interval. This finding shows that when selecting relay vehicles for a platoon, platoon vehicles should be favored over non-platoon vehicles.

In addition, Figure 8 shows that the average path repair time interval of a platoon is larger than 3 s. This value is higher (better) than that shown in Figure 5, which is only between 1 and 1.4 s. This finding confirms with one's expectation that IVC is more stable and useful in platoon communication.

6.2. Radio Penetration Rate

The second investigated system parameter is the radio penetration rate. In this investigation, we assume that all vehicles belonging to a platoon are equipped with a wireless radio, but not all the other vehicles are equipped with a wireless radio. The radio penetration rate is applied to those vehicles not belonging to a platoon. We varied the rate from 10 to 100% in our investigation.

In Figure 9, the results show that the average path lifetime increases as the radio penetration rate increases.

DOI: 10.1002/wcm

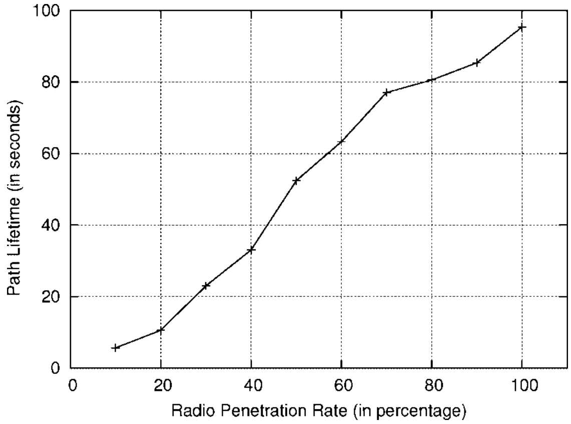

Fig. 9. The average path lifetime under different radio penetration rates.

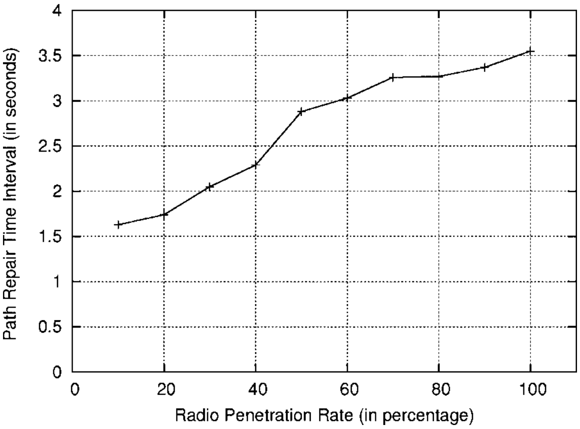

Fig. 10. The average path repair time interval under different radio penetration rates.

In Figure 10, the results show that the average path repair time interval increases as the radio penetration rate increases. Both results are expected and indicate that when the inter-vehicle distance of a platoon is larger than the wireless range, IVC relaying by the vehicles outside the platoon is important in maintaining the path between the leading and last vehicles of a platoon.

7. Discussion on Routing Protocol and Application Design for IVC Networks

In the first part of this paper, we have studied the lifetime and path breakage characteristics of routing paths in IVC networks and in platoons. Insights obtained from these results enable us to design routing protocols and applications that are more suitable for IVC networks.

For example, our results reveal that, in an IVC network on highways, the lifetime of most repairable routing paths are short and over 70% of these paths have a lifetime less than 10 s. However, in the studies of the platoon communication, the results show that longer path lifetimes can be obtained under the platoon-type IVC networks. In other words, platoon communication is more feasible in IVC networks.

As another example, our results show that the proposed policy, which prefers SDL links over ODL links during the path-selection phase of a routing protocol, indeed is effective in reducing the frequency of path breakage. Since implementing this policy is not difficult (e.g., by using an GPS receiver), routing protocols may consider using this path-selection policy when they are used in IVC networks. Besides, if a platoon vehicle can be chosen as a relay vehicle for its own platoon, it should be chosen with a high priority in the path-selection policy.

In addition, our results show that the path repair time interval of the paths in IVC networks is very short and on average a path needs to be repaired every 1.5 s. Even in the platoon communication, the interval does not increase significantly. These numbers indicate that it would be very difficult for any traditional unicast routing protocol to perform well on such networks. In the beginning of this paper, we presented that most routing protocols use HELLO-like messages to detect topology changes and to cope with unreliable wireless channels, and they usually need to receive/lose three consecutive such messages to add (delete) a vehicle into (from) a neighbor list. With the current HELLO interval of 1 s, this means that detecting a topology change would require at least 3 s. However, compared with the 1.5 s path repair time interval, very clearly, most current routing protocols will respond to path breakage too slowly in IVC networks.

One apparent approach is to decrease the HELLO interval from the current 1 s to, say, 0.1 s. However, this will increase the bandwidth overhead of these control messages by 10 times, which is undesirable. Another approach is to use location-based routing protocols (e.g., Reference [37]) to avoid leaving stale routing entries in highly mobile vehicles. However, because such protocols also rely on HELLO packets to maintain up-to-date neighbor lists, it is also difficult for them to quickly detect path birth, disappearance, and breakage in a fast-changing IVC network.

Another approach, which is used in our protocol, is not to use HELLO messages to detect topology changes, but instead use some forms of intelligent-flooding mechanism to reliably and efficiently forward

DOI: 10.1002/wcm

packets in an IVC network. In this approach, when a packet needs to be sent, it is flooded from the source to the destination vehicle with a mechanism to cancel redundant transmissions. Flooding provides the best reliability because it can use any path to reach the destination vehicle as long as one exists. Flooding also copes with fast-changing networks well because any path available can be used to deliver the packet. As such, unlike in a traditional routing protocol, the difficult task of maintaining and updating neighbor lists and routing states in a fast-changing IVC network can be eliminated.

In the second part, we will present the design and implementation of our intelligent flooding-based routing protocol for IVC networks and its performance in real-world IVC networks.

8. Protocol Architecture

We first present the protocol architecture on the sending, forwarding, and receiving nodes and the terms used in our protocol. The architecture of our protocol has two advantages. First, our protocol can transparently support all real-world applications without the need to modify them. Second, our protocol can be easily implemented on many operating systems and ported to different platforms. Due to these advantages, the NISSAN Taiwan automobile Inc. is working with us to integrate our protocol into its TOBE set-top boxes, which run WinCE and are standard equipments in some models of NISSAN cars.

We implemented our protocol as a user-level daemon program (called FloodRD) rather than a kernel module of the operating system. When the application generates packets and uses socket system calls to sends them out, we use a packet filtering and capturing mechanism provided on most operating systems (e.g., ipfirewall on FreeBSD and Netfilter on Linux) to capture them in the kernel and redirect them to FloodRD before they are passed to the network interface for transmission. After FloodRD processes them, they are directly passed to the network interface for transmission without being captured again. By this design, our FloodRD protocol processing is transparent to real-world applications and independent of operating system platforms.

Figure 11 shows the protocol architecture on the sending node for the Linux operating platform. It also illustrates the detailed processing for broadcasting a unicast packet: (1) a unicast application sends data into the kernel through a UDP or TCP socket for transmission. (2) The kernel prepends an IP header to the data to form a unicast packet for transmission. However, Netfilter captures and redirects the packet to the user-level FloodRD daemon. (3) FloodRD encapsulates the directed packet (called ‘raw packet’ on Linux) with a FloodRD-specific header. FloodRDs on other nodes will use the information in this header to decide whether they should further flood the packet. (4) FloodRD calls the sendto( ) socket system call to send the encapsulated packet into the kernel as data for transmission. The kernel prepends an IP header to the data as normal. Because the destination IP address provided to the sendto( ) socket system call is a broadcast IP address, the destination IP address in this header is a broadcast IP address. The kernel then passes the formed broadcast packet to the underlying network interface for transmission. Since Netfilter is configured in such a way that it captures and redirects only the packets generated by normal applications, the broadcast packets generated by FloodRD will not be captured and redirected to FloodRD again.

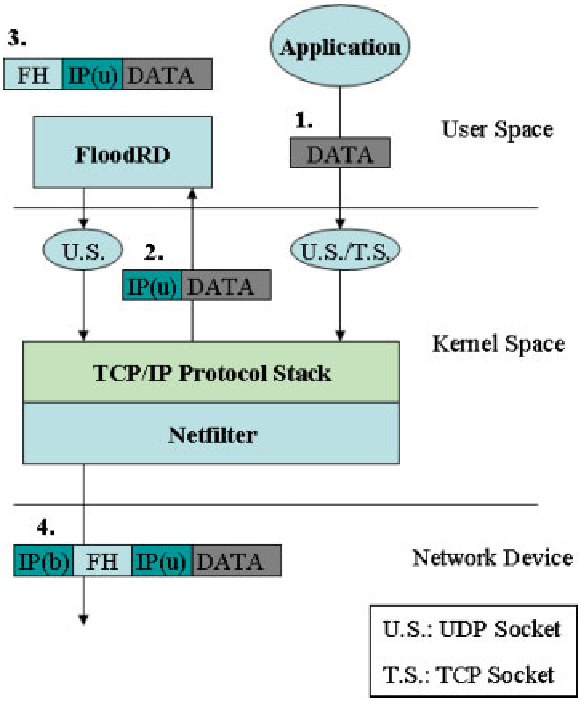

Fig. 11. The protocol architecture on the sending node.

Figure 12 shows the processing flow of a unicast packet when it is flooded on its way from the sending node, *via* the forwarding node(s), and to the destination node. The packet processing on the sending node has been explained above. When FloodRD on the sending node transmits a broadcast packet, a neighboring node overhearing the transmission will deliver the

DOI: 10.1002/wcm

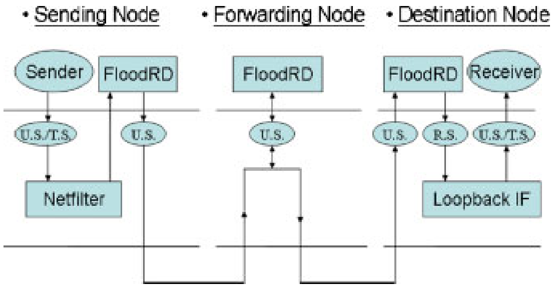

Fig. 12. The processing flow of a flooded packet.

packet to its FloodRD. The FloodRD first determines whether the raw packet has reached its final destination by comparing the destination IP address in the raw packet's header with its own IP address. If this node is not the packet's destination node, FloodRD may decide to drop the packet to avoid redundant transmissions or further forward it. In the latter case, like on the sending node, FloodRD will rebroadcast this raw packet. A packet may need to go through multiple forwarding nodes before reaching its destination node.

In case the packet has reached its destination node, FloodRD on the destination node simply sends the raw packet into the kernel through a raw socket. Because the destination IP address of the raw packet is the same as the IP address of the destination node, the raw packet is looped back by the loopback interface in the kernel as if it had just been received from a physical network interface. Using the normal processing, the data payload of the raw packet is enqueued into the receive queue of the socket on which the application is listening. The application then uses socket system calls (e.g., recvfrom( )) to read data from the socket.

9. Protocol Design

Although flooding provides high reliability and eliminates the need to install and update routing entries in an IVC network, it has two shortcomings. First, in the design of IEEE 802.11 (a/b/g) MAC protocols, a broadcast frame is not protected by ACK frames. In contrast, a unicast frame is protected by ACK frames, by which the sending node can detect frame losses and resend the lost frame for up to seven times. This difference makes broadcast frames less reliable than unicast frames at the MAC layer. Second, a large number of redundant packet transmissions may result due to flooding and these redundant packets will waste network bandwidth. These problems are addressed in our protocol design. In the following, we elaborate on each part of our protocol design.

9.1. Error Handling

Lacking an ACK protection mechanism at the MAC layer for broadcast frames can result in a poor performance at a higher layer in our design. Therefore, our protocol implements our own ACK and retransmission mechanisms for broadcast frames.

In our design, each transmitted broadcast packet should be acknowledged to indicate that either its destination node has received the packet or a forwarding node has received the packet. If the corresponding 'ACK' packet cannot be received within a period, which is a system parameter, the packet is rebroadcasted. Rebroadcasting a packet is repeated until a threshold is reached, which is also a system parameter.

To save the number of ACK packets, our protocol does not require a forwarding node to explicitly send back an ACK packet to the previous node that broadcasts the packet. Instead, the acknowledgment function is implicitly achieved by 'hearing' that at least one forwarding node has received and rebroadcasted the packet. 'Hearing' is possible because if some forwarding node rebroadcasts the packet, this means that it is in the transmission range of the previous node. Because a wireless link between two nodes is symmetric in most conditions, when the next forwarding node wirelessly rebroadcasts the packet, the previous node should be able to 'hear' (receive) the packet. To let FloodRD know whether a received packet is broadcasted by an upstream node or by a downstream forwarding node, a sequence number is used and carried in the FloodRD-specific header of the packet. This sequence number will be increased by one each time when the packet is forwarded.

Due to this design, the per-hop acknowledgment function is mostly achieved without ACK packet bandwidth overhead. In case the previous node does not hear the packet rebroadcasted by the next forwarding node (this situation may occur due to packet collisions or temporary link asymmetry), the previous node will rebroadcast the packet. In the case of the last hop, the destination node should explicitly send an ACK packet to the previous node because it has no need to further forward the packet.

9.2. Redundant Transmission Avoidance

Redundant transmissions are minimized by three schemes in FloodRD. The first scheme is the

DOI: 10.1002/wcm

band-flooding scheme proposed in Reference [42] to limit the scope of a flooding message to a narrow band between the source and destination nodes. Using the second scheme, a forwarding FloodRD will not rebroadcast the same packet more than once. In an IVC network, a packet may be cloned and each of these copies takes a different path to reach a forwarding FloodRD. Since each FloodRD associates each locally generated packet with a per-source-node different sequence number, a forwarding FloodRD can easily identify the cloned copies of a packet and drop them. Using the same scheme, FloodRD discards duplicated copies of a packet and delivers only one copy of the packet to the application on the destination node. Using the third scheme, a forwarding FloodRD will not rebroadcast a packet once it hears that some FloodRD has rebroadcasted the packet. In our protocol, FloodRD delays the transmission (i.e., forwarding) of a packet by a random delay time and if a FloodRD hears that another FloodRD has forwarded the packet, it will cancel its own packet transmission. Suppose that a FloodRD broadcasts a packet and its neighboring FloodRDs receive the packet. All of these neighboring FloodRDs will try to rebroadcast the packet to forward it. However, since only one FloodRD will first forward the packet and the rebroadcast is heard by other neighboring FloodRDs, these FloodRDs will not rebroadcast the same packet.

Consider the case of the last-hop forwarding. Because the packet has reached its destination node and there is no need to further forward the packet, all FloodRDs around the destination node should cancel their pending packet transmissions when they receive the ACK packet sent out by the destination node. To ensure that this ACK packet can reach these FloodRDs before their packet transmissions start, the lower bound of the random delay times should be larger than the time required for the destination node to send out an ACK packet. This lower bound is a system parameter and its value should be chosen based on measurements.

9.3. Sliding Window-based Forwarding

Unlike the MAC design of IEEE 802.11 (a/b/g), which requires an immediate ACK frame for the current transmitted data frame, in our protocol the 'ACK' packet (either an explicit ACK generated by the destination node or an implicit ACK indicated by the forwarding of the packet) need not be immediate. The IEEE 802.11 (a/b/g) MAC protocol uses a stop-and-wait protocol (i.e., the sliding window size is one frame per round-trip-time (RTT)) for its ACK mechanism. Although this protocol is simple and can be easily implemented, it suffers from poor performances when the RTTs between the transmitting and receiving nodes is large. Our protocol is implemented at the user level as a FloodRD daemon for portability reasons. However, this advantage comes with the cost that the RTT between two FloodRDs is larger than the RTT between two MAC modules in wireless network interfaces. To achieve a good performance, FloodRD adopts a sliding window-based ACK mechanism, which allows several packets to be transmitted successively before the ACK packet for the first packet has arrived.

9.4. Fast Retransmission

If a lost packet can only be retransmitted by the expiration of the retransmit timer, the performance will be poor. For this reason, FloodRD implements a TCP-like fast retransmission mechanism to quickly retransmit a lost packet.

To implement the fast retransmission mechanism, FloodRD attaches a locally generated per-node sequence number to each forwarded packet. By inspecting the sequence numbers carried in incoming ACK packets (either explicit or implicit ACKs), FloodRD views that a packet has been lost if its ACK packet has not come back but more than N ACK packets for its following packets have come back, where N is a system parameter. In this case, FloodRD will retransmit the lost packet immediately.

9.5. In-Order Forwarding and Delivery

In-order packet forwarding and delivery are important for higher layer protocols and applications to achieve good performances. For example, excessive out-of-order packets may cause a TCP receiver to send back cumulative ACK packets to the TCP sender, which will trigger TCP fast retransmission and TCP congestion control and reduce the current sending rate by a half. Our protocol needs to handle this problem because (1) packets generated by the same application may reach a forwarding FloodRD in the wrong order due to traversing on different paths in the network, and (2) FloodRD artificially gives different random delay times to each forwarded packet to cancel redundant transmissions. For these reasons, packets pending to be forwarded are sorted in the transmission queue based on their per-source-node sequence numbers rather than their initial random delay times. If a newly arriving packet (say, packet A) has a smaller sequence number than that of a packet already in the transmission queue

DOI: 10.1002/wcm

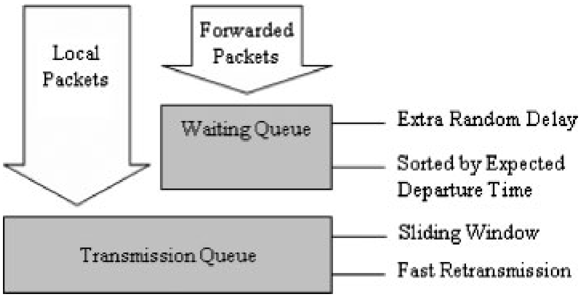

Fig. 13. The internal organization of FloodRD.

(say, packet B) but packet A is given a larger random delay time than that given to packet B, FloodRD will exchange their random delay times to make sure that packet A will be transmitted before packet B and the random delay effect is still maintained.

9.6. Overall Organization

Figure 13 depicts the internal organization of FloodRD. Locally generated packets are inserted to the transmission queue directly for transmission. Packets to be forwarded are put into the waiting queue waiting for their random delay timers to expire. Their positions in the waiting queue are adjusted by the in-order forwarding and delivery mechanism. When the random delay timer of a packet expires, the packet is moved to the transmission queue for transmission. Before moving to the transmission queue, packets in the waiting queue are very likely to be canceled due to the redundant transmission avoidance mechanism. Packets in the transmission queue are transmitted under the control of the sliding window-based forwarding mechanism. Lost packets are retransmitted under the control of the fast retransmission mechanism.

10. Implementation

This section presents the detailed implementation, configurations, and settings of our scheme running on Linux. The system parameters used in our scheme, their current values, and their effects are also presented.

10.1. Packet Filtering

We separate the network used by FloodRDs from the network used by real-world applications. The wireless network interface of each node is configured with two IP addresses using the ‘IP aliasing’ utility. One IP address is on the 1.0.1 subnet while the other IP address is on the 1.0.2 subnet. Choosing IP addresses for these interfaces can be arbitrarily done. The 1.0.1 subnet is used for normal applications to exchange their data. The 1.0.2 subnet, on the other hand, is used internally by FloodRDs to flood packets generated by applications. For example, if a FloodRD wants to broadcast a packet to its neighboring FloodRDs, it uses the 1.0.2.255 subnet broadcast address as the destination IP address of the packet.

Using two different subnets makes the protocol design clear and allows the packet filtering and capturing rule to be easily specified as ‘only capture and direct packets whose destination IP address is on the 1.0.1 subnet’. With this rule, a packet is captured and redirected to FloodRD only once on its sending node because FloodRD will encapsulate the packet with a new IP header that uses 1.0.2.255 as the destination address. Using a new destination address on a different subnet, a packet will not be captured and redirected infinitely on its sending node. In addition, broadcast packets that enter and leave forwarding nodes and broadcast packets that enter the destination node will not be captured and redirected.

10.2. Broadcasting Packets at a High Rate

Some IEEE 802.11 (a/b/g/) network interface cards (NICs) purposely slow down the transmission rate for broadcast packets to only 2 Mbps. Obviously, such NICs will severely limit the maximum performance of our scheme because FloodRDs broadcast packets to forward them. To overcome this problem, we changed the configuration of the driver for these NICs and allowed them to broadcast packets at the 11 Mbps high transmission rate.

10.3. Parameter Settings

Table I lists the names and descriptions of the system parameters used by FloodRD. The maximum allowable length of the transmission queue (RQ_MAX_LENGTH) affects packet loss rate, achieved throughput, and end-to-end packet delays. The size of the sliding window (RQ_SWINDOW_SIZE) determines how many packets can be successively transmitted per RTT to increase the forwarding throughput. Using a larger size may increase the forwarding throughput but at the cost of a higher degree of packet reordering. The threshold for triggering the fast retransmission mechanism (RQ_FAST_RETX_THRES) determines how many ACKs of following packets should be received before retransmitting the lost packet. Using

DOI: 10.1002/wcm

Table I. The parameter settings used by FloodRD.

Name	Description	Value
Parameters related to the transmission queue		
RQ_MAX_LENGTH	Max queue length	5 pkt
RQ_SWINDOW_SIZE	Sliding window size	2 pkt
RQ_FAST_RETX_THRES	Threshold for triggering fast retransmission	5 pkt
RQ_RETX_TIMEOUT	Max waiting time for ACK	80 ms
RQ_RETX_MAX_COUT	Max retransmission times	1
Parameters related to the waiting queue		
WA_FIXED_DELAY	Minimum random delay time	8 ms
WA_RDELAY_RANGE	Random delay time range	15 ms

a smaller value for the threshold enables FloodRD to quickly detect a packet loss but at the cost of more unnecessary retransmissions. The maximum waiting time for an ACK (RQ_RETX_TIMEOUT) determines when to retransmit a packet if the fast retransmission mechanism fails to quickly retransmit the packet. Using a smaller value may increase the forwarding throughput at the cost of more unnecessary retransmission. The maximum retransmission count for a packet (RQ_RETX_MAX_COUNT) determines how many times a packet can be retransmitted. Using a larger value can decrease the packet loss rate at a cost of higher end-to-end packet delays. The minimum random delay time (WA_FIXED_DELAY) is the lower bound of random delay times given to packets to be forwarded. The range of the random delay times (WA_RDELAY_RANGE) determines the maximum difference between the smallest random delay time and the largest random delay time. Using a larger value for this parameter can spread generated random delay times more widely, which allows FloodRDs to cancel more redundant transmissions but at the cost of higher end-to-end packet delays and a higher degree of packet reordering.

11. Performance Evaluation

To evaluate the performances of our scheme, we conducted a series of field trials on an elevated highway. Six vehicles were used in experiments. Each vehicle is equipped with an IBM A-model laptop computer with a PCMCIA IEEE 802.11b NIC. The NIC in each vehicle was connected with a 5dbi external antenna *via* a 1.5-m thin signal cable, which are shown in Figure 14. Using an external antenna can increase the effective wireless transmission range to about 200 m. (If the external antenna is not used, the effective wireless transmission range is only about 30 m due to the vehicle's shielding effect.) Besides, the elevated highway provides the environment with line-of-sight wireless channels. The operating platform we used is Linux 2.6.7 kernel. We used STG and RTG, which is a pair of TCP/UDP traffic generators provided in NCTUns [43], to generate TCP/UDP traffic in experiments. We used kernel-AODV module [44] and our FloodRD as the underlying routing protocols and compared their performances.

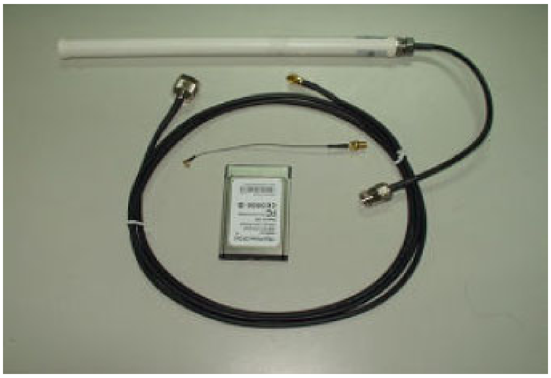

Fig. 14. The external antenna, cable, and wireless NICs used in field trial.

During the field trials, vehicles moved in the chain fashion. That is, they tried to move on the same lane with each vehicle following its previous vehicle. The first vehicle was the traffic source node and the last one was the traffic destination node. Two persons were required for each vehicle because one person needed to drive the vehicle while the other needed to operate the computer. Although each driver tried to maintain the distance between its own vehicle and the previous vehicle at a value close to and within the effective transmission range of the used wireless radios, the distances between vehicles unavoidably had to vary due to unexpected and dynamic traffic and road conditions. As such, the required hop count for a packet to traverse

DOI: 10.1002/wcm

Fig. 15. The vehicles were prepared to move onto a highway for field trials.

from the first to the last vehicles might have to vary over time during the field trials. (The possible hop count at any time can be 1, 2, 3, 4, or 5.) Figure 15 shows that the used six vehicles were prepared to move onto a highway for field trials.

We first evaluated the performances of AODV in this IVC network. During the experiments, each of which lasted about 10 min, we logged detailed AODV-relevant events so that we can analyze various statistics after the field trials. Table II shows the AODV results with greedy UDP traffic. From the logged events, we found that AODV is very unsuitable for IVC networks. This table shows that among all the paths that could be set up between the source and destination vehicles during the experiment, 81.04% of them are single-hop (i.e., the source and destination vehicles are so close that they can communicate with each other directly), 18.95% of them are two-hop, 0.01% of them are three-hop, and no paths whose hop counts are greater than 4 can be found in the log. These results show that AODV is unable to quickly find long routes, maintain long routes, and update long routes in an IVC network.

As for throughput results, the average UDP throughput is 190.84 KB/s for single-hop paths, 40.91 KB/s for two-hop paths, 0.4 KB/s for three-hop paths, etc. These bad throughputs are caused by high bit-error-rate (BER) on these paths, which are caused by AODV using the shortest path between the source and destination nodes. Note that although using the shortest path can reduce the number of packet forwarding in the network, the quality of the found path is generally the worst because the distance between two neighboring nodes on the path should be as long as possible. The greedy TCP throughputs under AODV are not shown here because the measured TCP throughputs are all very close to 0 KB/s. We found that on the fast-varying IVC network, because packets are excessively lost due to route failures under AODV, it was almost impossible for a TCP connection to finish its three-way handshaking procedure to establish a connection. Even if a TCP connection could be set up, its sending rate mostly dropped to 0 KB/s due to many packet retransmission timeouts.

Table II. The average UDP throughputs with respect to different path lengths under AODV.

Path length (hop count)	Throughput (KB/s)	Path ratio (%)
1	190.84	81.04
2	40.91	18.95
3	0.4	0.01
4	0	0

Table III. The average UDP throughputs with respect to different path lengths under FloodRD.

Path length (hop count)	Throughput (KB/s)	Path ratio (%)
1.0 ~ 1.5	473.56	13
1.6 ~ 2.5	127.69	8
2.6 ~ 3.5	73.68	63
3.6 ~ 4.5	55.25	16

Table III shows the average UDP throughputs under FloodRD. The path lengths are not classified into distinct integers. Instead, they are classified into different ranges. The reason is that in our protocol, packets generated by an application are not transmitted over a fixed unicast path. Instead, these packets may take different paths with different lengths to reach their common destination node. Due to this reason, in every 1 s, FloodRD computes the average hop count of the packets received in the last second and their average receiving throughput. From the packet ratio results, we see that FloodRD can dynamically and quickly exploit every available route to deliver a packet from the source to the destination nodes. From the average throughput results, we see that FloodRD outperforms AODV greatly. This is because it does not suffer from the same high BER problem as AODV. In FloodRD, although a packet may be cloned, canceled, dropped, or lost, some of its copies will be propagated from the source to the destination nodes over any reliable path(s).

Table IV shows the average TCP throughputs under FloodRD. Although the TCP throughputs under

DOI: 10.1002/wcm

Table IV. The average TCP throughputs with respect to different path lengths under FloodRD.

Path length (hop count)	Throughput (KB/s)	Path ratio (%)
1.0 ~ 1.5	202.72	32.85
1.6 ~ 2.5	60.81	54.05
2.6 ~ 3.5	10.92	12.89
3.6 ~ 4.5	5.95	0.21

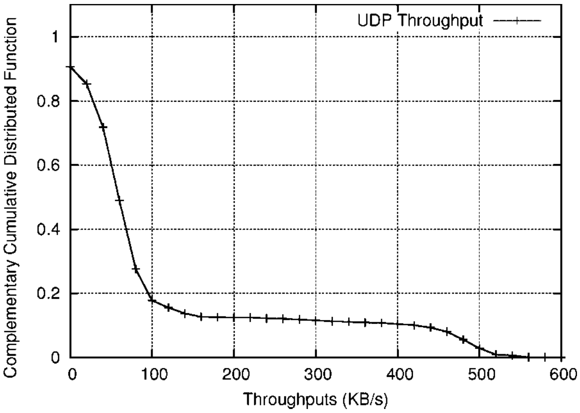

Fig. 16. The 1-CDF of UDP throughput over time under FloodRD during a field trial.

FloodRD are lower than the UDP throughputs under FloodRD, they are still much better than the TCP throughputs under AODV, which are close to 0 KB/s. Figure 16 shows the complementary CDF (1-CDF) of UDP throughput on this IVC network over time during a field trial while Figure 17 shows the 1-CDF of TCP throughput on this IVC network over time. From these figures, we see that more than 70% of the time the achieved UDP throughput under FloodRD (converged to about 40 KB/s when the path length is over 4 hops) suffices to support a low-rate video conferencing among a group of vehicles moving on the roads. In addition, the achieved TCP throughput under FloodRD suffices to support e-mail, FTP, and web applications that transfer a small volume of data.

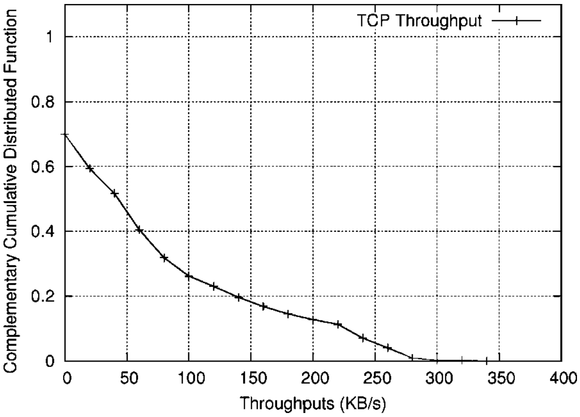

Fig. 17. The 1-CDF of TCP throughput over time under FloodRD during a field trial.

Table V. The average end-to-end packet delays with respect to different path lengths under FloodRD.

Path length (hop count)	End-to-end delay (ms)
1 ~ 2	9.60
2 ~ 3	41.12
3 ~ 4	74.99

We see that the achieved UDP throughput is higher and more stable than the achieved TCP throughput. As explained before, TCP performance is very sensitive to packet losses and the degree of packet reordering. On a fast-changing IVC network, achieving good TCP throughputs is not easy. We are investigating how to improve FloodRD to provide good TCP throughputs over an IVC network. Another research direction is to modify TCP so that the modified version can work well over an IVC network should TCP is really unsuitable for IVC networks.

Table V shows the average end-to-end packet delays under FloodRD. It shows that, as expected, if a packet needs to traverse more hops to reach its destination node, the end-to-end delay that it experiences will increase. The results show that under the current parameter settings, a packet will experience an extra forwarding delay of about 35 ms if it needs to traverse one more hop.

12. Other Applications

In addition to supporting non-real-time unicast applications such as e-mail, FTP, telnet, web, etc. among a group of vehicles, our FloodRD protocol supports three other important applications in ITS. In the following, we present each of them separately.

The first application is to support video conferencing between two vehicles moving on the roads. Video conferencing is a real-time application, which requires a minimum bandwidth to transport audio and video data in real time. If bandwidth is not enough, many packets will be dropped due to congestion, causing bad-quality

DOI: 10.1002/wcm

Fig. 18. A video conferencing was ongoing across a four-hop network under FloodRD.

video and audio. In addition, audio packets of a video conference should not experience too much delay in the network, otherwise the audio quality will be bad. Our previous results show that FloodRD can provide a stable UDP throughput of about 40 KB/s on an IVC network. At present, almost all video conferencing applications use UDP to transport audio and video packets. As such, if the video frame rate and the video frame size can be reduced so that the bandwidth required for the video conferencing application is less than 40 KB/s, FloodRD can support a video conference quite well. We have conducted this grade of video conferences in the field trials and found that FloodRD can support them well. Figure 18 shows the screenshot of a video conference supported by FloodRD on a four-hop IVC network. The messages and statistics shown on the left of the screen were generated by FloodRD.

The second application is to broadcast text, image, video, or audio among a group of vehicles moving on the roads. In ITS, timely broadcasting important traffic and road conditions to vehicles so that they can avoid congested or dangerous areas is important. For example, a toll station on the highway can use IVC to broadcast important information (e.g., a toll station is ahead, the toll station on this lane is temporarily closed, the toll fee, the road and traffic conditions ahead of the toll station, etc.) to all approaching vehicles so that the drivers can be alerted and prepared in advance. As another example, a camera can be set up at a crossroad and uses IVC to broadcast the image or video of the scene showing how many vehicles are waiting at the crossroad. Vehicles can use IVC to see the image or video of the scene and learn the current congestion level well before they arrive at the crossroad, thus having a chance to avoid the congested area. Our FloodRD protocol is well suited for this broadcast application because FloodRD uses flooding as its basic mechanism. To support this type of application, we just need to disable the redundant transmission avoidance mechanism of FloodRD. We have used FloodRD to broadcast audio/video among a group of vehicles in the field trials and the experimental results show that FloodRD works well for this type of application. Figure 19 shows an audio/video broadcasting under FloodRD. To simultaneously show the quality of received and displayed video on different computers, these computers are placed together in our laboratory. Note that although these computers are close and can communicate with each other directly, we purposely filtered out some frames at the MAC layers of these computers to create an artificial three-hop chain network topology.

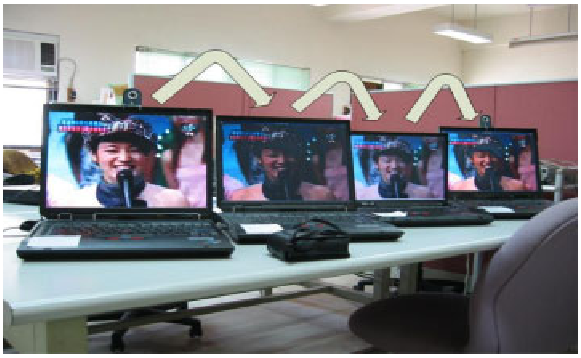

Fig. 19. An audio/video broadcasting was ongoing on a three-hop network under FloodRD.

The third application is to allow a vehicle on the roads to be connected to the Internet. This application enables a vehicle driver to send/receive e-mails and retrieve web pages from the Internet. Recently, IEEE 802.11 (a/b/g) WLAN hot-spots are becoming popular. As a result, it is becoming easier to find a WLAN hot-spot to connect to the Internet on the roads. Our FloodRD protocol design can easily allow a vehicle in an IVC network to connect to the Internet. To connect an IVC network using private 1.0.1.X and 1.0.2.X IP addresses with the Internet, we just need to run a FloodRD daemon on a gateway machine with two network interfaces. One interface is configured with a public IP address and connects the gateway to the public Internet. The other interface is an IEEE 802.11 (a/b/g) wireless network interface and connects the gateway to the private IVC network. Because the gateway is treated as one node in the IVC network, like all other nodes in the IVC network, its wireless interface is also configured with two IP addresses on the 1.0.1 and 1.0.2 subnets. Here, we assume that they are 1.0.1.254 and 1.0.2.254. To allow a vehicle (which uses private IP addresses) to communicate with

DOI: 10.1002/wcm

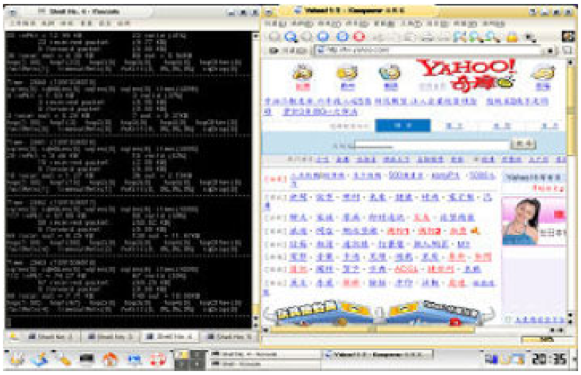

Fig. 20. FloodRD enables a vehicle to use IVC to fetch information from the Internet.

any host on the Internet, the FloodRD on the gateway is modified to also play the role of a network address translator (NAT) and the FloodRD on each vehicle is given the gateway's private IP address on the 1.0.2 subnet (say, 1.0.2.254) as its default gateway address. With these arrangements, when a vehicle sends a packet to a host on the Internet using that host's IP address (say, 140.113.17.71) as the destination IP address, the packet will be captured and redirected to the FloodRD as usual. The FloodRD first uses the IP-in-IP [45] protocol to encapsulate the packet with an IP header in which the gateway's IP address (1.0.2.254) is used as the destination IP address. It then encapsulates the packet with a FloodRD-specific header and then broadcasts the packet using 1.0.2.255 as the broadcast address. The packet is then flooded across the IVC network, reaches the gateway, and is finally received by the FloodRD on the gateway. Since the encapsulated packet is destined to the gateway, the FloodRD on the gateway will accept it. After seeing that the received packet is in the IP-in-IP format, the FloodRD strips the outer IP header off the packet to restore the original packet, which was generated by the application on the sending vehicle. At this time, the original packet has successfully traversed across the IVC network and reached its default gateway. From now on, the NAT module in the FloodRD can process the packet in the same way as it processes any packet that is delivered to it *via* a fixed wired network.

We have conducted field trials in our campus and found that FloodRD works well for this type of application. Figure 20 shows that a computer on a vehicle can connect to the Internet to fetch web pages from the Yahoo web site *via* FloodRD. The messages and statistics shown on the left of the screen are generated by FloodRD.

13. Future Work

We plan to test the performance of FloodRD using IEEE 802.11 (a/g) wireless LAN NICs. These NICs have higher bandwidths (e.g., 54 Mbps) than IEEE 802.11 (b) NICs (11 Mbps), which were used in our field trials. FloodRD with these NICs is expected be able to support higher quality video conferencing and broadcasting applications.

14. Conclusions

In this paper, we first studied the lifetime and breakage characteristics of routing paths in a simulated IVC network. We used a microscopic traffic simulator to generate more realistic vehicle mobility traces and then conducted simulations on these traces to derive the characteristics of routing paths. From the simulation results, we gained several insights into IVC networks. Based on these insights, we then designed and implemented an intelligent flooding-based routing protocol for small-scale IVC networks.

We conducted several field trials on highways to evaluate the performances of our protocol. Experimental results show that it is a practical routing protocol for small-scale IVC networks such as a platoon. First, it outperforms the famous AODV routing protocol greatly for unicast UDP and TCP data traffic. Second, it supports low-rate video conferencing applications. Third, it supports audio/video broadcasting applications. Finally, it enables a vehicle in an IVC network to connect to the Internet, by which the vehicle driver can send/receive e-mails and retrieve web information from the Internet. With these capabilities, the proposed routing protocol is practical and suitable for small-scale IVC networks (e.g., a platoon) on the roads.

Acknowledgment

This research was supported in part by MOE Program for promoting Academic Excellence of Universities under the grant number 91-E-FA06-4-4 and in part by NSC under the grant number 93-2213-E-009-089.

References

1. Perkins C, Royer E. Ad hoc on demand distance vector routing. In *Second IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

DOI: 10.1002/wcm

2. Turgut D, Das SK, Chatterjee M. Loggevity of routes in mobile ad hoc networks. In *IEEE Vehicular Technology Conference Spring 2001*, Greece, 6–9 May 2001; 2833–2837.
3. Yoon J, Liu M, Noble B. Random waypoint considered harmful. In *IEEE INFOCOM 2003*, March 2003.
4. Franz WJ, Hartenstein H, Bochow B. Internet on the road via inter-vehicle communications. In *Workshop der Informatik 2001: Mobile Communications over Wireless LAN: Research and Applications, Gemeinsame Jahrestagung der GI und OCG*, Wien, 26–29 September 2001 .
5. Sun M-T, Feng W-C, Lai T-H, Yamada K, Okada H, Fujimura K. GPS-based message broadcasting for inter-vehicle communication. In *2000 International Conference on Parallel Processing*; 279–287.
6. Morris R, Jannotti J, Kaashoek F, Li J, De Couto DSJ. Carnet: a scalable ad hoc wireless network system. In *9th ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System*, Kolding, Denmark, September 2000.
7. Da Chen Z, Kung HT, Vlah D. Ad hoc relay wireless networks over moving vehicles on highways. In *The ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001) Poster Paper*, October 2001.
8. Briesemeister L, Schafers L, Hommel G. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *IEEE Intelligent Vehicle Symposium*, October 2000; 522–527.
9. Briesemeister L, Hommel G. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *The First Annual Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc 2000)*, August 2000.
10. Yashiro T, Kondo T, Yagome H, Higuchi M, Matsushita Y. A network based on inter-vehicle communication. In *IEEE International Conference on Intelligent Vehicles*, 1993; 234–250.
11. Kosch T, Schwingenschloegl C, Ai L. Information dissemination in multihop inter-vehicle networks—adapting the ad-hoc on-demand distance vector routing protocol (AODV). In *IEEE International Conference on Intelligent Transportation Systems*, Singapore, 3–6 September 2002.
12. Fuessler H, Mauve M, Hartenstein H, Kaesemann M, Vollmer D. MobiCom poster: location-based routing for vehicular ad-hoc networks. In *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 7(1), January 2003; 47–49.
13. Lochert C, Hartenstein H, Tian J, Fuessler H, Herrmann D, Mauve M. A routing strategy for vehicular ad hoc networks in city environments. In *IEEE Intelligent Vehicles Symposium (IV2003)*, Columbus, OH, June 2003; 156–161.
14. Wang SY. On the effectiveness of distributing information among vehicles using inter-vehicle communication. In *IEEE ITSC'03 (International Conference on Intelligent Transportation Systems)*, ShangHai, China, 12–15 October 2003.
15. Wang SY. On the intermittence of routing paths in vehicle-formed mobile ad hoc networks on highways. *IEEE ITSC'04 (International Conference on Intelligent Transportation Systems)*, Washington DC, USA, 3–6 October 2004.
16. Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review* 1994; **24**(4): 234–244.
17. Johnson D, Maltz DA. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, lmielinski T, Korth HF (eds). Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996; 153–181, Chapter 5.
18. Ko YB, Vaidya NH. GeoTORA: a protocol for geocasting in mobile ad hoc networks. In *Network Protocols, 2000. Proceedings, 2000 International Conference*, Osaka Japan, 14–17 November 2000; 240–250.
19. Lee SJ, Gerla M. AODV-BR: backup routing in ad hoc networks. In *Wireless Communications and Networking Conference (WCNC) 2000 IEEE*, Vol. 3, Chicago, USA, 23–28 September 2000; 1311–1316.
20. Jiang M, Jan R. An efficient multiple paths routing protocol for ad-hoc networks. In *Information Networking, 2001, Proceedings, 15th International Conference*, Beppu City, Oita Japan, 31 January–2 February 2001; 544–549.
21. Das SR, Marina MK. On-demand multipath distance vector routing for ad hoc networks. In *IEEE ICNP 2001*, November 2001.
22. Ye Z, Krishnamurthy SV, Tripathi SK. A framework for reliable routing in mobile ad hoc networks. *INFOCOM 2003*, Vol. 1, 30 March–3 April 2003; 270–280.
23. Nasipuri A, Castaneda R, Das RR. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mobile Networks and Applications* 2001; **6**(4): 339–349.
24. Lee SJ, Gerla M. Split multipath routing with maximally disjoint paths in ad hoc networks. In *ICC 2001*, Vol. 10, Helsinki Finland; 3201–3205.
25. Sakurai Y, Katto J. AODV multipath extension using source route lists with optimized route establishment. In *IWWAN 2004*, University of Oulu, Finland, 31 May–3 June.
26. Obraczka K, Viswanath K. Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks* 2001; **7**(6): 627–634.
27. Dheap V, Munawar MA, Naik S, Ward PAS. Parameterized neighborhood-based flooding for ad hoc wireless networks. In *MILCOM 2003*, Vol. 2, 13–16 October 2003; 1048–1053.
28. Leng S, Zhang L, Yu LW, Tan CH. An efficient broadcast relay scheme for MANETs. *Computer Communications* 2004; **28**(5): 467–476.
29. Chandra R, Ramasubramanian V, Birman KP. Anonymous Gossip: improving multicast reliability in mobile ad-hoc networks. In *Distributed Computing Systems, 2001. 21st International Conference*, Mesa, AZ USA, 16–19 April 2001; 275–283.
30. Yi Y, Gerla M. Efficient flooding in ad hoc networks: a comparative performance study. In *ICC 2003*, Vol. 2, 28–30 May, Seattle Washington, USA; 1059–1063.
31. Hsu CS, Tseng YC. An efficient relaible broadcasting protocol for ad hoc networks. *IASTED Networks, Parallel and Distributed Processing, and Applications (NPDPA)*, 2002, Japan; 93–98.
32. Williams B, Camp T. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of MOBIHOC 2002*, Lausanne, Switzerland; 194–205.
33. Torrent-Moreno M, Jiang D, Hartenstein H. Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks. In *ACM VANET2004 Workshop*, Philadelphia, USA, 1 October 2004.
34. Korkmaz G, Ekici E, Ozguner F, Ozguner U. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *ACM VANET2004 Workshop*, Philadelphia, USA, 1 October 2004.
35. Namboodiri V, Agarwal M, Gao L. A study on the feasibility of mobile gateways for vehicular ad-hoc networks. In *ACM VANET2004 Workshop*, Philadelphia, USA, October 1 2004.
36. Wu H, Fujimoto R, Guensler R, Hunter M. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In *ACM VANET2004 Workshop*, Philadelphia, USA, October 1 2004.
37. Karp B, Kung HT. Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom 2000*, Boston, MA, USA, August, 2000.
38. VISSIM 3.60 User Manual, PTV Planung Transport Verkehr AG company.
39. Wiedemann. Simulation des Strabenverkehrsflusses. Schriftenreihe des Instituts fur Verkehrswesen der Universitat Karlsruhe, Heft 8, 1974.
40. Wiedemann. Modeling of RTI-Elements on Multi-Lane Roads. In *Advanced Telematics in Road Transport*, edited by the Comission of the European Community, DG XIII, Brussels, 1991.

DOI: 10.1002/wcm

41. The Network Simulator—ns-2, available at http://www.isi.edu/nsnam/ns
42. Wang SY. Reducing energy consumption caused by flooding messages in mobile ad hoc networks. *Computer Networks* 2003; **42**(1): 101–118.
43. Wang SY, Chou CL, Huang CH, *et al*. The design and implementation of the NCTUns 1.0 network simulator. *Computer Networks* 2003; **42**(2): 175–197.
44. The software is available at http://w3.antd.nist.gov/wctg/aodv_kernel/
45. Simpson W. IP in IP Tunneling, RFC 1853, October 1995.

Authors' Biographies

Shie-Yuan Wang is an Associate Professor of the Department of Computer Science at National Chiao Tung University, Taiwan. He received his Master and Ph.D. degree in Computer Science from Harvard University in 1997 and 1999, respectively. Before that, he received his bachelor degree in Computer Science from National Taiwan Normal University in 1990 and his Master degree in Computer Science from National Taiwan University in 1992. His research interests include wireless networks, Internet technologies, network simulations, and operating systems. He authors a network simulator, called NCTUns, which is now a famous tool widely used by people all over the world.

C. L. Chou currently is a Ph.D. candidate of the Department of Computer Science, National Chiao Tung University (NCTU), Taiwan. He received his bachelor degree and master degree in computer science from NCTU in 2000 and 2002, respectively.

Chih-Che Lin is a Ph.D. candidate of Department of Computer Science, National Chiao Tung University, Taiwan. He received his BS degree in computer science and information engineering from National Chiao Tung University, Taiwan, in 2002. He is a core team member of the NCTUns network simulator project. His current research interests include wireless networking, wireless mesh networks, vehicular networks, intelligent transportation systems, and network simulation.

DOI: 10.1002/wcm