Extending Drive-Thru Data Access by Vehicle-to-Vehicle Relay

Jing Zhao, Todd Arnold, Yang Zhang and Guohong Cao

Department of Computer Science & Engineering
Pennsylvania State University
University Park, PA 16803
{jizhao,tarnold,yangzhan,gcao}@cse.psu.edu

ABSTRACT

Recently, some researchers have performed extensive experiments to study the feasibility and performance of vehicle drive-thru access to roadside access points (APs). The experiments demonstrate that the duration of connectivity to the AP is limited. A drive-thru vehicle has an area of high signal strength near the AP, but experiences poor link quality when entering or exiting the AP coverage area. Since a vehicle spends a large portion of the connection time in this poor link quality area, the data throughput can be significantly reduced. This problem has been identified in several works, but a viable solution has yet to be identified. In this paper, we propose a vehicle-to-vehicle relay (V2VR) scheme which extends the service range of roadside APs and allows drive-thru vehicles to maintain high throughput within an extended range. Our solution is distributed and purely client-based, without any modification to the existing 802.11 APs. Through implementation and simulation, we demonstrate that the V2VR scheme can effectively extend the drive-thru access range and improve the network utilization for drive-thru vehicles.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design, Performance, Algorithms

Keywords

Vehicular network, relay, access point, roadside communication

1. INTRODUCTION

As mobile access becomes part of our daily life, there is a growing demand for accessing the Internet or information centers from vehicles. For example, access points (APs) can be deployed every few miles along the highway for users to download maps, traffic data, and multimedia files [13, 15, 14]. Vehicles can use APs to report real time traffic information and to assist other vehicles in avoiding traffic congestion. Although 3G networks or satellite techniques can be used to achieve this goal, roadside APs have the advantage of low cost, easy deployment, and high bandwidth.

To minimize the cost, the APs may be deployed several miles apart. In some cases, the APs being accessed are residential APs which are not an engineered network [2]. This should not present a problem for connecting to the Internet, as long as vehicles can frequently move through the AP coverage areas. However, unlike stationary users, vehicles move very quickly and only stay within an AP's coverage for a short time. Thus, one important challenge is to maximize the connection time and the amount of data transferred for the drive-thru vehicles.

In this paper we propose a relay-based solution to extend the service range of roadside APs. As a vehicle moves towards an AP, its signal quality with the AP may be poor. In order to extend its connection time and improve the throughput, the vehicle selects a vehicle geographically ahead of it to serve as a relay. The vehicle also selects a vehicle behind it to serve as a relay when it leaves the AP coverage area. The relay approach can improve the throughput and extend the AP coverage. When nodes close to the AP also need to access the AP, they may compete bandwidth with nodes asking for relay. However, this kind of problem may be worse without relay. This is because vehicles at the fringe of the AP coverage area compete with vehicles closer to the AP no matter there is a relay vehicle or not. Further, IEEE 802.11 is known to suffer from so-called *performance anomaly* [5, 7]. If the vehicles at fringe of the AP's coverage area directly access the AP using the lossy links, the AP is forced to adapt to a lower transmission rate to maintain good link quality, reducing the system throughput. The relay vehicle keeps the faraway vehicles from accessing the AP using one-hop poor link, which can help mitigate the performance anomaly problem for drive-thru vehicles.

Relay via multi-hop ad-hoc networks has already been extensively studied [10, 9, 12, 16]. The work in UCAN [10] focuses on finding a routing path from a mobile device to other mobile devices with better cellular network coverage. The work of rDCF [16] focuses on using a MAC layer based approach to relay, which is more efficient than establishing a relay path at the network layer. However, all these previous schemes cannot be easily applied to drive-thru vehicles be-

VANET'08, September 15, 2008, San Francisco, California, USA.

cause vehicle mobility can cause frequent interruptions of the relay link, which can significantly degrade the throughput. To the best of our knowledge, this is the first work to study wireless relay for vehicle drive-thru data access from both theoretical and empirical perspectives. We study the car following characteristics in the traffic and design a stochastic model to find reliable links among vehicles. We also develop a viable prototype to identify and resolve the implementation issues. Through simulation, we demonstrate that our solution well handles high vehicle mobility, and allows drive-thru vehicles to maintain high throughput in the extended AP coverage area.

The reset of the paper is as follows. Section II gives the motivation of the work. Section III presents our relay scheme. In Section IV, we present the experimental implementations. Section V evaluates the performance of the proposed relay scheme, and Section VI concludes the paper.

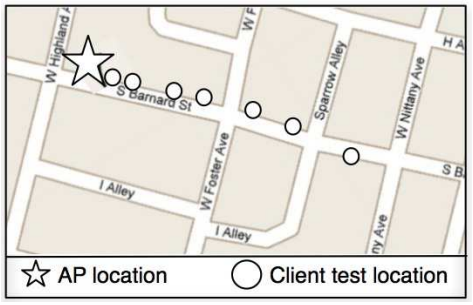
2. MOTIVATIONS

In order to verify our hypothesis that one can use ad-hoc relay to increase the AP access time, we conducted a simple experiment. Figure 1(a) and (b) show the experiment setup. Our testbed consisted of a Linksys wrt54GL wireless router as the AP, a Linux server, a PowerBook G4 laptop as relay and a Linux laptop as the client. The AP was mounted on the roof of a two-story house on the roadside in a residential area. The client laptop communicated with either the relay laptop or the AP using a Prism chipset based Orinoco 802.11b wireless card. The relay laptop communicated with both the client laptop and the AP using an Airport 802.11a/b/g wireless interface and a Netgear USB card. The server was connected to the AP through a high speed cable. In order to test the two hop AP access performance, we configured the relay laptop to forward packets from the client laptop destined to the server by masquerading with IP tables. We also manually changed the routing table of the client laptop, adding the relay laptop as the default gateway so that the client laptop sent all generated traffic to the relay laptop. We incrementally moved the client away from the AP location, keeping the relay laptop equidistant between the AP and the client.

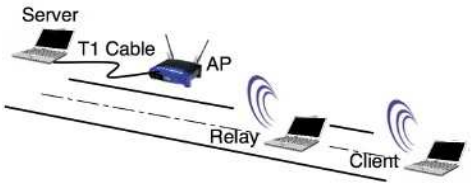
We performed two experiments. The first experiment tested what percentage of the AP beacons could be received at different distances from the AP. The second experiment tested the UDP throughput between the client and the server using Netperf [8].

Figure 2(a) shows the results of our first experiment. As can be seen, the beacons could be reliably received (above 90%) when the client stayed within 60 meters of the AP. Outside this range, the delivery ratio had an abrupt drop, and the link became lossy. The results also showed that even though the client was fairly far away from the AP (e.g. 100 meters), it could still receive 22% of the beacons.

Our second experiment (Figure 2(b)) compared one-hop and two-hop UDP throughput at different distances. The points in the figure are the throughput values we collected. We also generated a line of best fit based on the collected data by the regression method. In the figure, the one-hop UDP throughput was consistently above 2.8 Mbps within the 60-meter range from the AP. Outside this range, an abrupt throughput drop could be seen, and the throughput quickly dropped below 0.5 Mbps. This rapid throughput drop beyond a certain distance was reported in other works



(a) Experiment environment



(b) Experiment device setup

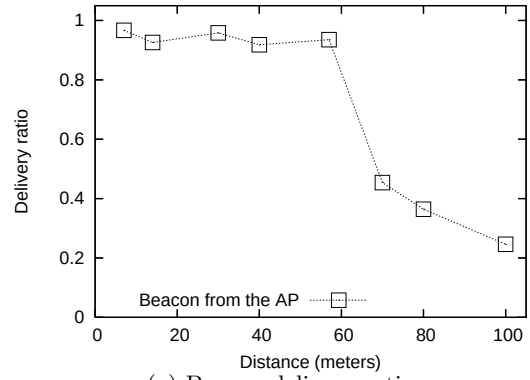
Figure 1: Experiment setup.

[3, 6, 11]. By using the two-hop approach, the throughput decrease was much smaller compared to the single hop approach. As shown in the figure, the two-hop approach could still achieve good throughput (close to 2 Mbps) outside the 60-meter range of the AP, while the single hop approach was below 0.5 Mbps. It also extended the AP access range with fairly high and stable throughput (above 1.8 Mbps) to 100 meters, almost twice as far as where stable throughput could be achieved by the one-hop approach. However, the throughput of the one-hop approach was better than that two-hop approach within the 60-meter range. We also find that the location where the abrupt drop of the one-hop throughput coincided with the location where the massive beacon loss was observed.

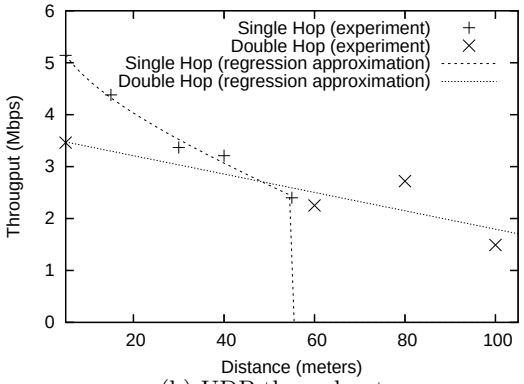
The simple experiment demonstrates the potential of using a relay to improve the client access range and throughput. First, within a specific range from the AP, stable and high single hop throughput can be achieved. Outside of this range, the throughput will be significantly reduced. However, the two hop relay solution can still achieve high and stable throughput. Therefore, finding a relay node inside this range may greatly improve the AP access duration. Second, the abrupt throughput drop is related to the beacon delivery ratio. Thus, we can use the beacon delivery information to effectively determine the location to switch between direct access and using relay.

In the above experiment, the client laptop is placed stationary at each testing location, which is different from the real drive-thru scenario. However, it has been shown in the existing field experiments [2, 3, 11] that vehicle speed has little effect on the packet delivery rate and throughput at a given distance to the AP. So we believe our experiment, though based on static scenario, still provides an adequate base for further proposing our scheme.

3. THE VEHICLE-TO-VEHICLE RELAY (V2VR) SCHEME



(a) Beacon delivery ratio



(b) UDP throughput

Figure 2: Beacon delivery ratio and UDP throughput at different distance from the AP.

3.1 Assumptions

We assume that GPS is available in every vehicle to report the location. We also assume that each vehicle is equipped with two 802.11 wireless interfaces: one is pre-configured to infrastructure mode and the other is in ad hoc mode. We believe multiple wireless interfaces will be common for vehicles, since many applications require the infrastructure mode and the ad hoc mode to both be active and the cost of a wireless interface is low. For instance, when a roadside AP disseminates road hazard emergency information, vehicles rely on the infrastructure mode to receive the data. At the same time, ad hoc mode may be used to propagate emergency warning to drivers behind a vehicle (or incident) to avoid multi-car collisions.

3.2 Scheme Overview

Figure 3 depicts the basic idea of our V2VR scheme. V2VR allows vehicles to establish the relay connection before entering the AP coverage area. When a vehicle wants to extend its AP access time on the road, it tries to find two vehicles as proxies; one in front of itself (*forward proxy*) and the other behind itself (*backward proxy*). The vehicle can establish a connection with a proxy through the wireless interface in ad hoc mode. After a link to the proxy is established, the vehicle may use the forward proxy to relay its traffic to an AP before entering the AP coverage or at the fringe of the AP coverage area. Similarly, when a vehicle is leaving the AP coverage area and its connection with the AP becomes poor, it uses the backward proxy to relay traffic. To avoid the overhead of frequently changing proxies, the vehicle attempts to find vehicles with similar mobility to keep the relay connection for an extended period of time.

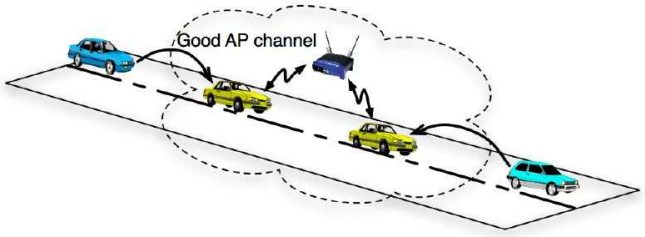


Figure 3: Access AP through a relay.

Our proposed V2VR scheme has three steps. In the first step, a client vehicle[1] finds a group of proxy vehicles with similar mobility as itself, and registers with them. The client vehicle will only ask vehicles in this group to act as relays because its connection with the selected group is more reliable than other vehicles. The second step occurs when a proxy vehicle enters an AP's coverage area; the proxy will select one client registered with itself to relay its traffic. This is called the *forward relay scheme*. Forward relay enables a client vehicle to access the AP earlier and improves the client's data throughput. It also allows a client to seamlessly switch to the infrastructure mode once a reliable connection to an AP is established. The third step occurs when a client vehicle is leaving the AP's coverage area and the connection becomes poor. The client finds a proxy behind itself to relay traffic backward to the AP, which is called the *backward relay scheme*.

A vehicle can determine when it has a reliable connection to the AP based on the beacon message delivery ratio. In 802.11, the beacon rate is specified in the header of the beacon packet. Thus, a vehicle can simply count the number of beacon packet it hears during a time frame, and compute the delivery ratio. The area with high beacon delivery ratio implies a vehicle can reliably communicate with the AP, e.g. within 60 meters in our experiment (see Figure 2(b)). The vehicle can also find the location where the delivery ratio has an abrupt drop, which is called the *AP optimal access range (AOR)*.

It is well recognized in traffic flow theory that vehicles in the traffic are generally grouped in a "platoon pattern", with a surge of vehicles, followed by a gap in traffic [4]. Vehicles in a group generally move with similar velocities which are determined by the leading vehicle. In a short distance (e.g. 400-600 meters where an AP is accessible), few vehicles in a group would dramatically change their velocities, leaving the platoon group in a consistent state. In the V2VR scheme, we propose a stochastic model to explore the above properties so that a vehicle can find a proxy in its platoon group with good probability, and the connection can remain stable for a short period of time.

[1]A vehicle attempting to upload or download data through an AP is called a *client vehicle*, or a *client*; a vehicle that is willing to relay traffic for other client vehicle is called a *proxy vehicle*, or a *a proxy*.

3.3 Step 1: A Stochastic Model for Finding A Reliable Proxy Group

A client attempts to find proxies to relay its traffic. It is not practical for a client to rely on one proxy for an extended period of time due to mobility. However, frequently searching for new proxies should also be avoided because the increased overhead can offset the benefit of relay. Thus, in V2VR, a client attempts to establish reliable one-hop links with selected proxies and ensure the links do not break within a reasonable period of time, e.g. several minutes. We define this time period as *lease time* and the selected proxies as *reliable proxies*. Clients search for new, reliable proxies only after lease time expires. A reliable proxy is expected to have similar mobility as the client vehicle; vehicles may move at different speeds and in different lanes. Mixed with acceleration and deceleration, it may be difficult for the client to find a proxy with matching mobility, or to even measure the mobility similarities. Fortunately, vehicle movement on the roadway is not completely random and we are particularly interested in exploring the movement similarities for a group of vehicles in a "platoon pattern". The movement of a vehicle in a platoon group is restricted by the vehicle ahead and behind itself, as well as the roadway speed limit. Thus, it has some well defined movement characteristics, which can help to predict the relative location of two vehicles in a platoon group after a period of time.

In this section, we propose a stochastic model to predict the relative distance of two vehicles in a platoon group based on the vehicle's past movement profile. The result can help the client find a group of proxies with similar mobility, so that the links between them are more likely to be stable and last longer.

Table 1: Notations

n	-	Number of lanes.
v_i	-	Speed of vehicles in Lane i.
(i, j)	-	Pairing state. It represents that the client and proxy vehicle are in Lane i and Lane j respectively.
$a_{i,j}$	-	Rate at which the client vehicle change from Lane i to Lane j.
$b_{i,j}$	-	Rate at which the proxy vehicle change from Lane i to Lane j.
t	-	Lease time. Time period when a reliable proxy group is valid.

For vehicles moving in a platoon, each vehicle is followed by another in a lane. A roadway may have multiple lanes, and we assume that all the vehicles in the same lane travel at an identical *lane speed*, and different lanes have different lane speeds. In order to pass other vehicles, a vehicle must change to a faster lane, and the speed is determined by the lane speed.

The mobility profile of a vehicle is defined by the lane changing rates of the vehicle. Let $a_{i,j}$ denote the lane changing rate of vehicle a from Lane i to Lane j. On a n-lane roadway, the mobility profile of the vehicle a can be formally defined by Formula 1.

$$M = \{a_{i,j} | 1 \leq i, j \leq n\} \tag{1}$$

The lane changing rate over a given period of time can be easily computed by the vehicle itself. A vehicle, say vehicle a, only needs to record the time it spent in the previous lane, every time it moves to another lane. Then it can easily compute its average time spent in Lane i before moving to j. The reciprocal of the average time yields the value of $a_{i,j}$. If we assume the time of a vehicle staying in a lane is exponential, the process of vehicle a moving from Lane i to Lane j can be modeled as a Poisson process with mean of $a_{i,j}$. Since vehicles can only move to the adjacent lanes, $a_{i,j}$ satisfies $a_{i,j} = 0, \forall 0 \leq i, j \leq n, |i - j| > 1$.

To study the relative movement of a client and a proxy vehicle, we propose a Markov Chain Model. The state of the Markov Chain is defined as (i, j), where i and j represent the lane number of the client vehicle and the proxy respectively. The amount of time the process spends in state (i, j) before making a transition is exponential with a mean $1/u_{(i,j)}$. $u_{(i,j)}$ is related with the rates of the client and proxy moving to their adjacent lanes. Let $a_{i,j}$ and $b_{i,j}$ represent the lane changing rate of the client and proxy respectively, $u_{(i,j)}$ can be computed by Formula 2.

$$u_{(i,j)} = a_{i,i-1} + a_{i,i+1} + b_{j,j-1} + b_{j,j+1} \tag{2}$$

When the process leaves state (i, j), the probability it enters the next state (p, q) is computed by Formula 3

$$P_{(i,j)\to(p,q)} = \begin{cases} \frac{a_{ip}}{u_{(i,j)}}, & |i - p| = 1, j = q \\ \frac{b_{jq}}{u_{(i,j)}}, & |j - q| = 1, i = p \\ 0, & \text{else} \end{cases} \tag{3}$$

Given this Markov Chain model, we can compute the proportion of time spent in any state (i, j), denoted as $P_{(i,j)}$. When $i = j$, the client and proxy are moving at the same lane, so they don't change their distances over time. However, when $i \neq j$, the client and the proxy are moving at different lanes. Larger $P_{(i,j)}$ implies that the two vehicles would develop more relative displacement over time. On the other hand, the difference between v_i and v_j also affects how much relative displacement it will generate after a period of time. Therefore, we use Formula 4 to estimate the relative displacement between the client and the proxy after a period of time t.

$$\Delta d = \sum_{1 \leq i,j \leq n} P(i, j) \cdot (v_j - v_i) \cdot t \tag{4}$$

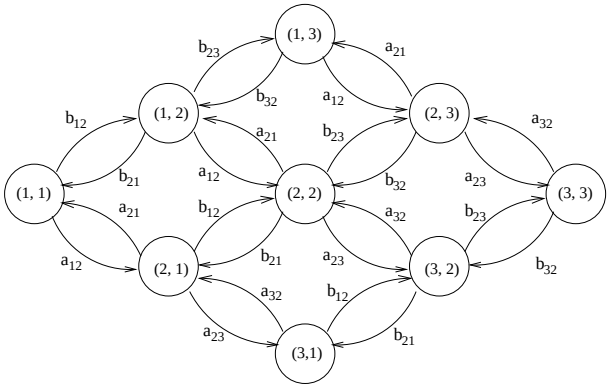


Figure 4: A Markov model for three-lane roadway.

In order to better present the Markov model, without loss of generality, we show an example of the model on a three-lane roadway. Figure 4 shows the Markov chain for the lane

changing states of the two vehicle, and Figure 5 shows the transition matrix of the Markov chain.

$$
\begin{array}{c}
 \\
(1,1) \\
(1,2) \\
(1,3) \\
(2,1) \\
(2,2) \\
\vdots \\
(3,3)
\end{array}
\begin{array}{c}
\begin{array}{ccccccc}
(1,1) & (1,2) & (1,3) & (2,1) & (2,2) & \cdots & (3,3)
\end{array} \\
\begin{pmatrix}
0 & \frac{b_{12}}{u_{11}} & 0 & \frac{a_{12}}{u_{11}} & 0 & \cdots & 0 \\
\frac{b_{21}}{u_{12}} & 0 & \frac{b_{23}}{u_{12}} & 0 & \frac{a_{12}}{u_{12}} & \cdots & 0 \\
0 & \frac{b_{32}}{u_{13}} & 0 & 0 & 0 & \cdots & 0 \\
\frac{a_{21}}{u_{21}} & 0 & 0 & 0 & \frac{b_{12}}{u_{21}} & \cdots & 0 \\
0 & \frac{a_{21}}{u_{22}} & 0 & \frac{b_{21}}{u_{22}} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0
\end{pmatrix}
\end{array}
$$

Figure 5: Transition matrix for three-lane Markov model.

To obtain the proportion of time in each state $P_{(i,j)}$, we compute the equivalent limiting probability, so we can derive the following equation system.

State $(1,1)$: $P_{(1,1)}(a_{12}+b_{12}) = P_{(1,2)}b_{21} + P_{(2,1)}a_{21}$

State $(1,2)$: $P_{(1,2)}(a_{12}+b_{21}+b_{23}) = P_{(1,1)b_{12}} + P_{(1,3)}b_{32} + P_{(2,2)}a_{21}$

State $(1,3)$: $P_{(1,3)}(a_{12}+b_{32}) = P_{(1,2)}b_{23} + P_{(2,3)}a_{21}$

$$\vdots$$

State $(3,3)$: $P_{(3,3)}(a_{32}+b_{32}) = P_{(2,3)}a_(23) + P_{(3,2)}b_{32}$

$$\sum_{1 \le i,j \le 3} P(i,j) = 1$$

Solving the above equation system yields the proportion of time in each state (i,j), so the client can compute the displacement between the proxy and itself during the lease time. In the Markov chain model, a n-lane roadway generates $n \times n$ states. In the real world, since n cannot be very large, the computational overhead of this model is very small.

To determine whether the proxy is a reliable proxy, the client needs to consider the relative mobility, the distance to the proxy and the signal quality. Thus, a client only considers a vehicle as a reliable proxy when Formula 5 is satisfied.

$$d + \sum_{1 \le i,j \le n} P(i,j) \cdot (v_j - v_i) \cdot t < R \qquad (5)$$

In Formula 5, d represents the current distance between the client and the proxy, t represent the lease time, and R represents the effective range of the wireless link in ad hoc mode.

With the criteria to determine a reliable proxy, we present the protocol for finding a reliable proxy group. Figure 6 shows the protocol of selecting a reliable proxy. The proxy selection algorithm requires client vehicles to request the mobility profile and location of the potential proxies. After a client vehicle broadcasts a *Proxy Discover* message, every proxy vehicle will reply with a *Proxy Advertise* message, which is a 3-tuple *(location, mobility profile, transmission power)*. The client uses the transmission power from the proxy to compute the SNR by comparing with the receiving power of the message. If the SNR is below a pre-specified value, the proxy is not considered. The location and mobility profile from the proxy are used to determine whether it is reliable, using the proposed Markov chain model. All the proxies which satisfy Formula 5 are recorded as reliable proxies by the client, while reliable forward proxies and reliable backward proxies are differentiated based on their geographic locations relative to the client.

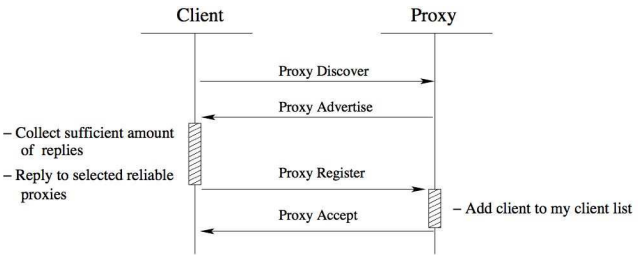


Figure 6: Select a relay proxy.

After the client determines the available and reliable forward and backward proxies, it sends a *Proxy Register* message to each of these proxies. The message also informs the proxies of the client's lease time and location. The proxy then stores an entry for the registered client and is ready to forward traffic for the client when it meets an AP, and it sends a *Proxy Accept* message back to the client.

Thus, before meeting any APs, a client can find and register with its reliable forward proxies and backward proxies. These proxies are bound to the client and may help the client forward data traffic before the lease time expires. Since one proxy may be selected by multiple clients as their reliable proxy, a proxy also records a list of registered clients. When the proxy meets an AP, it will select one client from the list to help forward data traffic. The binding between clients and reliable proxies are many to many instead of one to one. Thus, the first step of V2VR helps a client to determine a set of *possible* proxies before entering the AP coverage, but does not force the client to connect one specific proxy. The V2VR scheme put off the selection of the actual relay proxy until the client and proxies move into a specific AP. At that time, the real relay link between a client and a proxy can be established quickly with low overhead, and with the help of the existing client and proxy many-to-many binding. We believe this is the best way to achieve high reliability, efficiency and low overhead relay. Next, we will discuss how to establish a real relay link between a client and a specific proxy.

3.4 Step 2: Forward Relay

3.4.1 Establishing the forward relay connection

When a proxy vehicle receives the first beacon from an AP, it attempts to connect to a client which has already registered to it. Because of mobility, a registered client may not still have a good link to the proxy or may not be able to communicate. Thus, the proxy needs to poll the registered client and see if the link is still alive and whether the link quality still meets the requirements. If not, the proxy attempts to serve the next registered client. The registered clients are first sorted as a list: initially the client that the proxy has successfully served in the last AP coverage always comes first. The other registered clients are ordered by the client registration time, i.e., first registered client is selected first. The proxy polls the clients by the order of the sorted

list. After a registered client is selected, the proxy sends a *Forward Relay Available* message to it and waits a short period of time (50 milliseconds in our simulation) for the client to respond. When the client receives the message from the proxy, it first checks whether the link quality is good enough by checking the SNR with a pre-specified threshold. The client agrees to use the proxy only when it satisfies the link quality. If so, it replies with a *Forward Relay Request* message. Otherwise it does not send any message to the proxy. After the proxy receives the Forward Relay Request message, it replies the client with a *Forward Relay Confirm* message to confirm the establishment of the forward relay connection. If the client is not within the proxy communication range or the client decides not to use the proxy, the proxy will not receive any response during the time interval. Then the proxy will put the selected client to the tail of the sorted list and check the next client in the sorted list.

All proxies which have not connected to any clients turn their ad hoc mode interfaces into promiscuous mode. When they overhear the *Forward Relay Confirm* message, they know a client-proxy connection has already been established, so if they have the same client in their own registered client lists, they will mark this client and do not poll this client when they search their relay clients. This is very effective in reducing the redundant client polling, because the proxies behind the connected proxy may otherwise check the same client again.

Note that there is enough time for the proxy to poll multiple clients because there is delay between the proxy receives the first beacon (where it is usually at the fringe) and it gets good link quality to the AP. The proxy has enough time to poll several clients before its link with the AP becomes stable.

3.4.2 Connecting to an AP

When a forward proxy determines it has a good channel quality with an AP, it sends a *Forward Relay Start* message to its connected client. Upon receiving the message, the client can start to transfer data with the AP through the proxy relay. While using the ad hoc interface to transmitting data, the client also keeps its infrastructure interface active, listening for beacon messages from the AP and checking the SNR. When its infrastructure interface gets good link quality, it switches to use the infrastructure interface to directly connect to the AP.

3.5 Step 3: Backward Relay

The backward relay is proposed to connect a client back to the AP through a proxy when the client is moving away from an AP and the direct link becomes poor. Different from the forward relay scheme, it is the client who selects the relay proxy from its reliable backward relay group in the backward relay scheme. The selection procedure is similar though. The client needs to poll its reliable backward proxies one by one until it finds a proxy with good enough link quality. The client also sorts its reliable backward proxies and stores them in a sorted list; initially the backward proxy that successfully served itself in the last AP coverage always comes first. The other reliable backward proxies are ordered by their distances to the client at the time of client relay registration in a decreasing order, i.e., the farthest proxy is polled first. The proxy poll the proxies by the order of the sorted list. After the client selects a backward proxy, it sends a *Backward Relay Request* message to the proxy and waits a short period of time (0.1 seconds in our simulation) for the response. If the proxy has good link quality to both the AP and the client (by measure SNR) and is ready for backward relay when receiving Backward Relay Request message, it replies the client with a *Back Relay Confirm* message to confirm the establishment of the backward relay connection. The proxy may not be within the client communication range or it is already busy forwarding traffic, e.g. it is a forward proxy for another client. Thus the proxy will not reply to the client. Without receiving the response from the proxy within the specified time interval, the client will put the selected proxy to the tail of the sorted list and check the availability of the next proxy in the sorted list. After successfully finding a backward proxy, the client will switch from the direct AP link to the two-hop relay link to access the AP.

Figure 7 compares the AP access process under the traditional 802.11 standard and the proposed V2VR scheme.

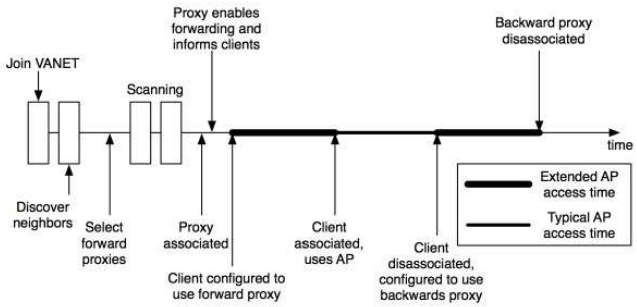


Figure 7: The traditional 802.11 and V2VR AP access timeline.

4. IMPLEMENTATION

4.1 Background

To test the feasibility of our ideas, we implemented a simple relay scenario. The implementation was focused on the study of the operations and status transitions which are required to setup the relay path on the fly. The implementation was also designed to test how quickly a proxy could enable packet forwarding and inform a client of its connection to an AP. We also wanted to determine how quickly a client could begin using the forward proxy once informed of the proxy's connection with the AP.

Our implementation was performed in a static environment. We used two laptops running Linux, kernel v2.4.5, with two NICs in each laptop. Our AP was a Linksys WRT54GL running DD-WRTv23 as the operating system, with a third Linux laptop to act as the server connected directly to the AP via a high-speed cable. Our packet capture was performed on an Apple Powerbook using Ethereal.

In order to relay packets, the ability to manipulate the relevant fields in packet headers when we forwarding packets from the VANET interface to the infrastructure interface was required. IP tables provide the capability to manipulate packet headers, known as masquerading. It also provides the rules for forwarding packets and can be enabled or disabled on the fly. To listen for our messages, we used the PCAP

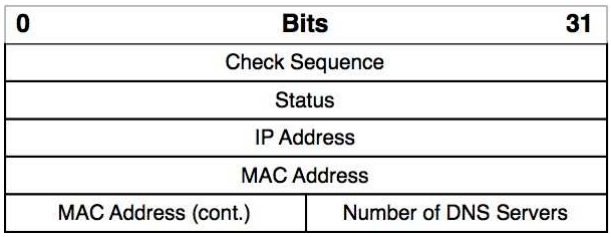


Figure 8: Hello packet format. The packet contains the minimum amount of information required for a trailing vehicle to use another vehicle to forward its traffic.

library to capture packets. All of our system information was retrieved and configured with the ioctl system call. We use MAC broadcasts as our message format to minimize overhead and vehicles do not necessarily know the address of their neighbors.

To format our hello packet, which covers the roles of the proxy request and selection messages as well as the forward relay available message, we had to determine the information required for a vehicle to use another as a proxy. The minimum information is the potential proxy's IP address, so the trail vehicle knows where to send its packets. To reduce the network overhead, we included the potential proxy's MAC address to eliminate the need for ARP requests. Unless the potential proxy also decides to act as a DNS server, the DNS Server IP address(es) provided by the AP are also required for accessing the Internet from the trail vehicle. If the potential proxy is associated with the AP, it stores the DNS server information in the file /etc/resolv.conf. The same file is manipulated by the client after it receives a hello packet containing DNS information. The resulting packet format is seen in Figure 8.

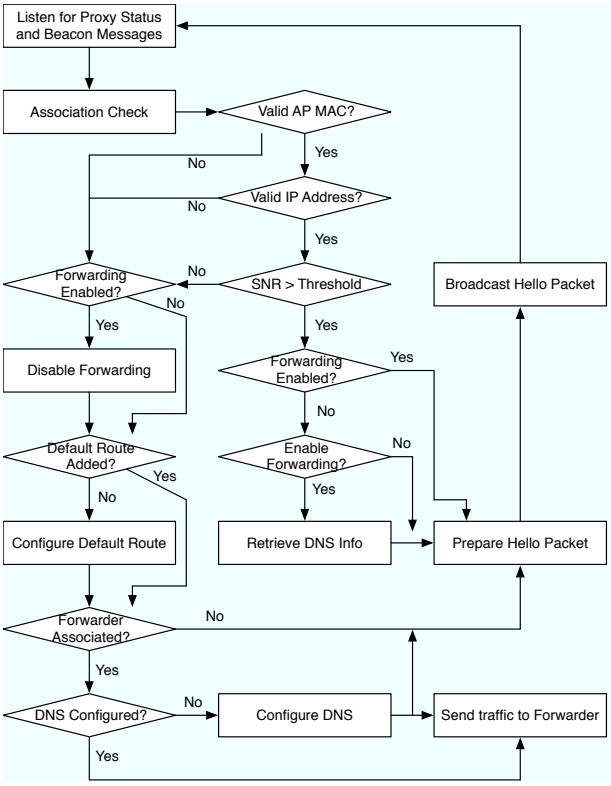


Figure 9: The decision making process and information flow of the implementation. We define being associated as the traditional definition, plus having a valid IP address and an acceptable SNR. If the vehicle can verify that it is associated, it will use the AP. Otherwise, it will use the vehicle that it chose to act as its forwarder for passing traffic to the AP.

4.2 Information Flow

In order to exchange information, the two vehicles need to communicate with one another. One of our assumptions is that each vehicle is equipped with two interfaces among which one is used strictly for vehicle to vehicle communication. Over this interface we are able to establish forwarding relationships.

Since a vehicle can act as either a proxy or a client at any time, it must determine which role it needs to play. Figure 9 is a visual representation of the entire decision making process for a vehicle. Given that the vehicles can communicate, the vehicle will also have selected its proxy as discussed in Section 3. The first step in the process is to determine whether or not the client needs to use the proxy, which requires determining whether or not the vehicle is associated with an AP. We extend the notion of association to include having a valid IP address and a SNR above our reliability threshold. If any of these three requirements are not met, the vehicle considers itself not associated.

If the vehicle is associated, it must determine whether or not it wants to act as a proxy for other vehicles. If forwarding is not enabled and the node is willing to serve as a proxy, it enables masquerading and packet forwarding. The vehicle also removes the old default route, the one pointing at its proxy. If it does not choose to act as a relay, it simply sends the hello packet without a status change. If it has already enabled forwarding previously, that means it has already been through the decision making process and there are no changes required. The final step, which only applies when a vehicle is associated and is willing to act as a forwarder, is to retrieve the number of DNS servers the AP provided for inclusion in the hello packet.

If the vehicle fails any of the association checks, it must use the information provided by its selected proxy to configure a default route to forward traffic through the proxy. If the client has already configured the default route, nothing is changed and the node only prepares its hello packet for transmission. However, if the node has not configured itself to leverage the proxy, it must configure a default route and install an entry in its ARP cache.

The node also checks to see whether or not the proxy is able to forward traffic at this time. If so, then the node will use the DNS information contained in the hello packet to configure its own DNS entries. If the proxy is not associated or DNS is already configured, then nothing is done except packet transmission.

In preparing the packet, a node includes its own status of being an eligible proxy, its own IP, MAC, and if associated, DNS server information.

To perform these checks and transmit hello packets, we used a 100 ms interval. The interval is equivalent to the

Packet #	Elapsed Time	Source	Destination	Protocol	Bytes	Information
1	0.000000	00:1a:73:37:8a:d1	ff:ff:ff:ff:ff:ff	IEEE 802.3	130	Hello packet
2	0.105106	00:1b:2f:3e:3d:b9	00:18:39:ea:5f:04	IEEE 802.11	98	Authentication[Malformed Packet]
3	0.105987	00:18:39:ea:5f:04	00:1b:2f:3e:3d:b9	IEEE 802.11	106	Authentication[Malformed Packet]
4	0.107098	00:1b:2f:3e:3d:b9	00:18:39:ea:5f:04	IEEE 802.11	118	Association Request
5	0.108577	00:18:39:ea:5f:04	00:1b:2f:3e:3d:b9	IEEE 802.11	122	Association Response[Malformed Packet]
6	0.110374	00:1b:2f:3e:3d:b9	00:18:39:ea:5f:04	IEEE 802.11	110	Probe Request
7	0.111642	00:18:39:ea:5f:04	00:1b:2f:3e:3d:b9	IEEE 802.11	143	Probe Response
8	0.219166	0.0.0.0	255.255.255.255	DHCP	428	DHCP Discover - Transaction ID 0x20f19200
9	0.278245	00:18:39:ea:5f:04	Broadcast	ARP	128	Who has 192.168.1.106? Tell 192.168.1.1
10	1.297803	00:18:39:ea:5f:04	Broadcast	ARP	128	Who has 192.168.1.106? Tell 192.168.1.1
11	2.218863	00:18:39:ea:5f:04	Broadcast	ARP	128	Who has 192.168.1.106? Tell 192.168.1.1
12	2.714642	192.168.1.1	192.168.1.106	DHCP	428	DHCP Offer - Transaction ID 0x20f19200
13	2.716917	0.0.0.0	255.255.255.255	DHCP	428	DHCP Request - Transaction ID 0x20f19200
14	2.720883	192.168.1.1	192.168.1.106	DHCP	428	DHCP ACK - Transaction ID 0x20f19200
15	2.995471	00:1a:73:37:8a:d1	ff:ff:ff:ff:ff:ff	IEEE 802.3	130	Hello packet
16	3.094725	00:1a:73:37:8a:d1	ff:ff:ff:ff:ff:ff	IEEE 802.3	134	Hello packet
17	3.460062	10.0.2.3	192.168.1.121	ICMP	184	Echo (ping) request
18	3.674821	00:1b:2f:3e:3d:b9	ff:ff:ff:ff:ff:ff	ARP	128	Who has 192.168.1.121? Tell 192.168.1.106
19	3.680323	00:0d:9d:85:b9:09	00:1b:2f:3e:3d:b9	ARP	146	192.168.1.121 is at 00:0d:9d:85:b9:09
20	3.680698	192.168.1.106	192.168.1.121	ICMP	184	Echo (ping) request
21	3.681044	192.168.1.121	192.168.1.106	ICMP	184	Echo (ping) reply
22	3.681346	192.168.1.121	10.0.2.3	ICMP	184	Echo (ping) reply

Figure 10: The sequence of events for a client to use a proxy. The proxy broadcasts its hello packets and goes through a typical association process with an AP (1 – 15). Once association is complete, the proxy's next broadcast includes DNS information (16). After receiving status change and DNS address, the client uses the proxy to relay traffic (17 – 22).

default beacon interval for our AP, which allows the node time to attempt to associate, or go through the association process, and to update neighbors in a timely fashion. In our full scheme, we would not perform all of these checks at a specific interval, except the association check. This would cut down on transmission overhead by reducing unchanged updates.

4.3 Experimental Results

The two laptops were configured to communicated with one another prior to either associating with the AP. The lead vehicle sends hello packets to the trail vehicle, who selects its forwarder based on the fact that it only hears hello packets from one neighbor. The hello packets are packets 1, 15, and 16 in Figure 10. The packets were sent at 100 millisecond intervals, but were removed to save space. To simulate movement, the potential proxy associates with the AP and requests an IP address, as seen in packets 2 – 14 in Figure 10. Once the lead vehicle fully associates with the AP, it must note its status change, include the DNS server information in its hello packet, and send the information to the trail vehicle. The inclusion of a single 4-byte DNS server can be seen in the size increase from packet 15 to 16. It also shows that the lead vehicle spends less than 100 milliseconds (between two hello packets 15 and 16) to configure itself and get ready to forward traffic. Upon receiving the change in status information, the trail vehicle is able to install the DNS information and can begin accessing the Internet via the lead vehicle in under four-tenths of a second; the trail vehicle must write to its /etc/resolv.conf file, which accounts for the delay in sending its first ICMP request. The success of the forwarding can be seen in the ping sequence of packets 17 – 22. The trail vehicle sends a ping request to its gateway, which acts as the intermediary for the request and response.

The next step is to associate the trail vehicle with the AP and for it to halt sending traffic via the lead vehicle. The trail vehicle must go through the same association process. Once the trail vehicle is fully associated with the AP, it removes the old default route that pointed at the lead vehicle and is able to immediately begin forwarding traffic directly through the AP.

5. PERFORMANCE EVALUATIONS

5.1 Simulation Setup

To evaluate the performance of the proposed scheme, we implemented it in ns-2 (version 2.30) to compare it with the generic 802.11-based AP access scheme (No Relay). We simulate a simple scenario where 150 vehicles pass an AP on an one-way road, traveling at speed of 45 miles/hour (20 m/s). The vehicles move into the road randomly, following an exponential distribution with a mean of 0.1 to 2 vehicles/second; equivalent to a mean inter-vehicle space of 10 to 200 meters. Among these 150 vehicles, 10%-50% vehicles are randomly selected to generate network traffic. Each selected vehicle initiates an FTP session and sends data to the AP via TCP (Reno) immediately after associated with the AP. The vehicle continues sending data until it loses connection with the AP.

Table 2: Simulation Setup

Parameter	Value
Simulation area	$2000m \times 500m$
Number of vehicles	150
Vehicle coming rate	0.1 - 2 vehicles/second
Vehicle velocity	45 miles/hour
# of TCP senders	15 - 75
Multi-NIC setting	2 radios/5 channels
Transport layer module	TCP Reno
MAC layer module	802.11b
Bandwidth	2 Mbps
Data packet size	1440 byte
AP beacon interval	100 ms
AP optimal range (AOR)	60 meters

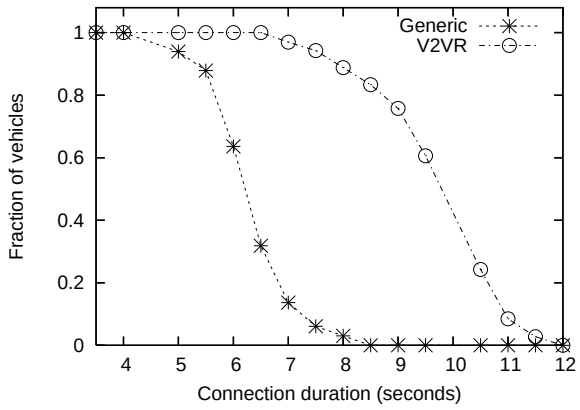


Figure 11: Complementary CDF of the connection duration. The mean duration for the no-relay scheme and the relay scheme is 6.06 and 9.68 respectively.

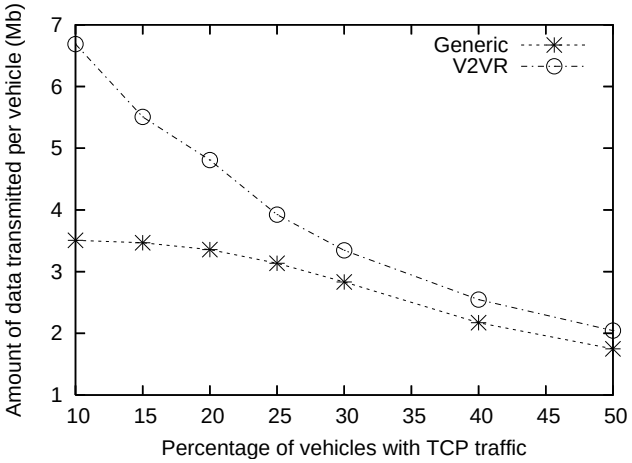
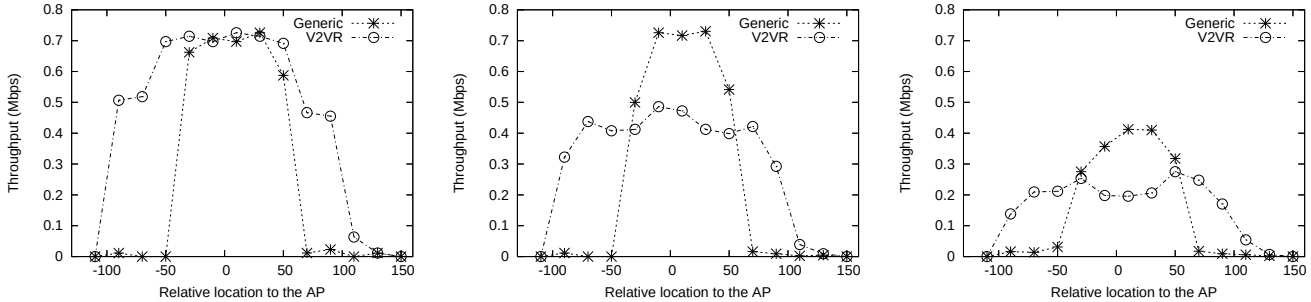


Figure 12: Average TCP traffic transmitted per vehicle



(a) 10 % clients (b) 25 % clients (c) 50 % clients

Figure 13: Vehicle TCP data transmission throughput at different distances.

To emulate the link quality of a real environment, the simulation parameters are setup based on our experimental results. First, we tune the transmission power level in the simulator to provide 110 meter transmission range, matching our experimental results where no beacon packet is received farther than 110 meters. Second, the packet loss ratio at different distance from the AP is derived from our experimental results as shown in Figure 2, and we set the AP access range (AOR) to 60 meters. In our scheme, each vehicle has two NICs operating at different channels. Since the standard ns-2 does not support multi-channel, we add the two-NIC with multi-channel functionality based on the techniques provided in [1]. All the simulation parameters are shown in Table 2.

5.2 Simulation Results

5.2.1 AP Connection Duration

Figure 11 shows the complementary CDF of the connection duration between the AP and drive-thru client vehicles. The connection time is taken as the whole period during which a vehicle's TCP packet can be successfully delivered to the AP. In 802.11, the connection starts when a vehicle completes its association with the AP. When the vehicle receives the last TCP ACK message from the AP, the connection ends. In V2VR, if a proxy is found, the connection for the client starts when it successfully establishes a relay path to the AP, i.e., receiving the Forward Relay Confirm message from the proxy; otherwise, it waits until association finishes. As shown in the figure, V2VR significantly extends the connection time for moving vehicles. The average connection duration of our relay scheme is 9.68 second, which is 60% longer than that of the no-relay scheme (6.06 s). The extended connection time is obtained by both forward and backward relay. From Figure 11, we can see that the connection duration in the no-relay scheme is clustered around 6-7 seconds. During this time, the vehicle should be able to move 120-140 meters when traveling at 20 m/s. This implies that even if the vehicles can receive the AP beacon outside the AOR, they have little time to make use of the direct link with the AP. The connection durations of our relay scheme are more evenly distributed, since they rely on the opportunistic connection with the proxy vehicles.

5.2.2 Amount of Data Delivered

Figure 12 shows the average amount of data delivered by each vehicle as the number of vehicles with TCP traffic increases. When the data traffic going through the AP is light, our relay scheme can deliver much more data than the non-relay scheme, e.g. 90% more data when 10% of the vehicles are sending TCP traffic. This is due to the extended connection time and the improved link quality.

When more vehicles are sending data, less amount of data can be delivered by each vehicle in both schemes because all vehicles share the bandwidth. As the number of TCP senders increase, the improvement on the amount of data delivered between the two schemes becomes less pronounced, and will eventually converge. These results are not surprising when considering the fact that when the AP is saturated it cannot support more traffic even if vehicles are connected to the AP with our relay scheme.

5.2.3 Data Transmission Throughput

Figure 13 shows the TCP throughput that a client vehicle can obtain at different distances from the AP. In Figure 13, the location of the AP is taken as the reference point 0. The negative location value represents the area where the client moves towards the AP but has not reached it yet, while the positive location represents where the client has passed the AP. The figure shows the results when 10%, 25% and 50% of the vehicles are uploading/downloading data in (a), (b) and (c) respectively.

The results show V2VR can extend the stable throughput connection to the AP regardless of the client density, but it is more effective when the density is low. When a high percentage of the vehicles are simultaneously transferring data, the clients accessing the AP through relays compete for use of the channel with the one hop clients. Each client obtains lower throughput, which may offset the benefit of the extended connection time. When the client percentage is low, V2VR allows the client to obtain stable throughput much earlier.

We also notice that the stable connection area of 802.11 is skewed in the direction of vehicles leaving the APs range. The reason is that a client starts to send TCP data packets immediately after it associates with the AP, but its link quality is often still poor. We see frequent TCP timeouts and the TCP slow start is launched. When a client achieves good link quality, it is unable to achieve a high throughput due to TCP slow start. Since the link quality is only good in a limited time duration, 802.11 is not efficient in making use of the bandwidth.

6. CONCLUSIONS

In this paper we proposed a relay-based solution (called V2VR) to extend the service range of roadside APs. When the link quality between a drive-thru vehicle and the AP is poor, a relay with good link quality to the vehicle and the AP is chosen to improve the performance. We designed techniques to select forward and backward proxies based on the mobility pattern of the vehicle. We also developed a viable prototype to address the implementation issues. Experimental results and simulation results show that a significant number of vehicles can transmit much more data with our relay scheme than without relay.

7. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant CNS-0721479.

8. REFERENCES

[1] R. Agüero and J. P. Campo. Adding Multiple Interface Support in NS-2, January 2007.

[2] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *Proc. MOBICOM*, Los Angeles, CA, September 2006.

[3] R. Gass, J. Scott, and C. Diot. Measurements of In-Motion 802.11 Networking. In *Proceedings of the 7th IEEE Workshop on Mobile Computing Systems and Applications*, 2006.

[4] D. Gerlough and M. Huber. Traffic Flow Theory - A Monograph. *Special Report 165, Transporation Reseaerch Board*, 1975.

[5] D. Hadaller, S. Keshav, and T. Brecht. MV-MAX: Improving Wireless Infrastructure Access for Multi-Vehicular Communication. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, 2006.

[6] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *Proc. ACM MobiSys*, 2007.

[7] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proc. IEEE INFOCOM*, pages 836–843 vol.2, 2003.

[8] R. Jones. Netperf: A Network Performance Benchmark. Information Networks Division, Hewlett-Packard Company, February 1996.

[9] S. Lee, S. Banerjee, and B. Bhattacharjee. The Case for a Multi-hop Wireless Local Area Network. In *Proc IEEE INFOCOM*, 2004.

[10] H. Luo, R. Ramjee, P. Sinha, L. Li and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *Proc. ACM MobiCom*, 2003.

[11] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for "Automobile" Users. In *Proceedings of INFOCOM'04*, 2004.

[12] H. Wu, C. Qiao, S. De and O. Tonguz. Integrated cellular and Ad Hoc relaying systems: iCAR. *IEEE journal on selected areas in communications (JSAC)*, pages 2105–2115, Oct. 2001.

[13] Y. Zhang, J. Zhao, and G. Cao. On Scheduling Vehicle-Roadside Data Access. In *Proc. ACM VANET*, September 2007.

[14] J. Zhao and G. Cao. VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, May 2008.

[15] J. Zhao, Y. Zhang, and G. Cao. Data Pouring and Buffering on The Road: A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 56(6), Nov. 2007.

[16] H. Zhu and G. Cao. rDCF: A Relay-enabled Medium Access Control Protocol for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 5(9):1201–1204, September 2006.