

ROADRUNNER: INFRASTRUCTURE-LESS VEHICULAR CONGESTION CONTROL

Jason H. Gao, Li-Shiuan Peh

Massachusetts Institute of Technology
Computer Science & Artificial Intelligence Laboratory
32 Vassar St 32-G785, Cambridge, MA 02139, USA
jasongao@mit.edu, peh@csail.mit.edu

ABSTRACT

RoadRunner is an in-vehicle app for traffic congestion control without costly roadside infrastructure, instead judiciously harnessing vehicle-to-vehicle communications, cellular connectivity, and onboard computation and sensing to enable large-scale traffic congestion control at higher penetration and finer granularity than previously possible. RoadRunner limits the number of vehicles in a congested region or road by requiring each to possess a token for entry. Tokens can circulate and be reused among multiple vehicles as vehicles move between regions. We built RoadRunner as an Android app utilizing LTE, 802.11p, and 802.11n radios, deployed it on 10 vehicles, and measured cellular access reductions of up to 84% and response time improvements of up to 80%. In a microscopic agent-based traffic simulator, RoadRunner achieved travel speed improvements of up to 7.7% over an industry-strength electronic road pricing system.

Keywords: connected vehicles, personal devices, electronic tolling, congestion management, vehicular networks, VANET

INTRODUCTION

Traffic congestion is a widespread problem affecting road transportation infrastructure in many cities, and is expected to increase in severity. In 2005, congestion resulted in 4.2 billion hours of travel delay and 2.9 billion gallons of wasted fuel in the United States [30]. One widely studied approach to reducing congestion is road pricing, a monetary policy to disincentivize drivers from entering tolled regions. Road pricing has traditionally been implemented through manned toll booths but electronic toll collection systems are now common [2].

The Electronic Road Pricing (ERP) system deployed in Singapore in 1998 was the first in the world to apply electronic road pricing for congestion control of a large downtown area. It uses dedicated short-range radio communications (DSRC) to detect and collect tolls from vehicles passing under physical gantries on roads leading to heavily congested areas. Prices change throughout the course of a day [17, 19, 31]. In the United States, several open road high-speed tolling systems have been deployed: FasTrak in California (1993), SunPass in Florida (1999), and the Northeast's E-ZPass (1991) [16]. These systems use windshield-mounted radio transponders to communicate with physical gantries as vehicles drive by.

Congestion can also be controlled through regulatory or non-monetary policies that directly limit the number of vehicles that may drive on a road, known as road-space rationing. Prior studies in transportation research [14, 25] have shown that road-space rationing can improve road capacity when used solely or in conjunction with road pricing. However, deployments of road-space

rationing around the world [1] are mostly manual, policy-driven implementations.

In Singapore, the Area Licensing Scheme [27], predecessor to ERP, required vehicles to purchase and display a paper license before entering a restricted zone (RZ). There was a fixed quota on licenses and ALS was manually enforced by officers at RZ boundaries. Now, a quota on the total number of cars in Singapore is enforced through the Certificate of Entitlement (COE) system which requires a COE to be purchased before a car can be driven in Singapore. The COE lasts for 10 years and is auction priced, with an average price of S\$70K-90K in December 2012. Beijing implemented a temporary road-space rationing scheme by restricting even and odd license plate numbers on alternate days for three months prior to the 2008 Olympic Games. A slightly modified policy was implemented permanently following the successful three-month trial [37]. London similarly enforced road-space rationing for the 2012 Olympic Games.

All above-mentioned systems require the deployment of costly physical roadside infrastructure such as gantries, tollbooths or enforcement stations and personnel, and/or specialized in-vehicle devices. Thus, deployment of congestion control tends to be limited to a few large controlled regions within cities, each covering a wide swath of roads, making it very costly to re-define regions.

A congestion control system that does not require the setup of new physical infrastructure can address these downsides of existing systems, and enable widespread deployment of congestion control across entire cities, at the fine granularity of specific roads, permitting flexible definition of regions and quotas for more responsive policies. In fact, Singapore recently released a call to companies for proposing systems for the next-generation ERP that is to be GPS-based [9], with field trials currently underway.

In this paper, we propose, implement, and evaluate RoadRunner, an infrastructure-less congestion control system, with our prototype running on Android smartphones. Smartphones are widely adopted in many cities, with penetration reaching 50.4% [7] and 70% [6] in the U.S. and Singapore respectively. Phones can be readily plugged into vehicles, with car manufacturers providing smartphone docks on the dashboard [15], enabling seamless connectivity to substantial energy, driver-friendly interfaces, vehicle information, and vehicular communications such as DSRC [11, 32].

In the near future, RoadRunner could also be deployed on every vehicle via in-vehicle units (IVUs) rather than smartphones. All vehicles in Singapore are required to install an IVU equipped with DSRC for communications with ERP gantries, and the next generation of IVUs being field-tested in Singapore (ERP), Germany (simTD), and France (scoreF) will include GPS and 802.11p DSRC radios [8], enabling pervasive V2V communications, sensing, and computation.

Using smartphones enables an already widespread infrastructure-less solution to congestion control, but presents additional challenges. First, as smartphones and other mobile connected devices continue to proliferate, demand for cellular bandwidth is expected to exceed available capacity by 2014 [4]. The increased throttling and cost of 3G/4G data plans and phasing out of unlimited data plans are clear symptoms of increasing bandwidth pressure on mobile data networks [5]. A phone-based infrastructure-less system permits the extension of congestion control to all roads across an entire city, but a conventional client-server implementation will lead to millions of vehicles communicating through the cellular network to servers running and policing congestion control, creating intense bandwidth pressure on already overloaded cellular networks. Second, phone-based congestion control needs to swiftly respond to drivers so they can adapt their routing appropriately. A conventional client-server implementation which relies on the cellular data network may experience long and unpredictable latencies, especially when the network is heavily loaded in dense

areas [20, 29], and face difficulties meeting the real-time requirements of congestion control.

RoadRunner tackles the above challenges with a *distributed* congestion control system that offloads computing to nearby in-vehicle phones, leveraging vehicle-to-vehicle networking via ad-hoc WiFi and DSRC to ease the cellular bandwidth pressure and improve real-time response latencies. RoadRunner is a decentralized mobile phone app for vehicles to reserve places on roads in a transportation network. The system distributes tokens to vehicles as permission for their entry into regions or roads, and records infractions and/or enforces fines for violations of congestion control policies. Our prototype provides the driver with turn-by-turn voice directions just as in existing satellite navigation systems; A driver only has to enter a destination upon starting a trip, and RoadRunner automatically negotiates tokens and routing in the background. If a token cannot be obtained, RoadRunner notifies the driver of a change in route one intersection prior to the new route branch.

Our deployments on 10 vehicles show that RoadRunner improves mean system response times by 80% when coupled with DSRC radios for V2V communications, and reduces cellular data accesses by 84% compared to a traditional client-server implementation that only utilizes the cellular network. Our experiments also show that today’s smartphone GPS receivers have sufficient accuracy, that, when combined with buffer zones between regions, enable accurate identification of controlled regions entries and exits. Our simulation results (Section 6) show that RoadRunner can enable infrastructure-less congestion control on a large scale and improve travel speed over an existing sophisticated electronic road pricing scheme that varies tolls at different times of the day. By forgoing a charge, RoadRunner lowers costs for drivers compared to road pricing, making congestion control policies more palatable to the general public, permitting widespread deployment.

DESIGN

At a high-level, the goal of congestion control is to ensure that there are not too many vehicles on a particular segment of road at any one time. RoadRunner is an electronic token-based reservation system: vehicles must possess a corresponding *token* to drive on a specific road segment (*token*), analogous to road-space rationing.

Regions and a quota of tokens, provided by a central server, are pre-defined by the transportation authorities. Vehicles may not create or duplicate tokens, ensuring an upper bound on the number of vehicles in a region. Tokens can expire, which helps ensure that lost tokens are effectively reset and do not impede the operation of the system over a long period. If a vehicle in the region does not have a valid token, the system logs a violation and enforces a penalty, which could be a fine or reported infraction.

When a driver steps into a vehicle and begins a trip, RoadRunner determines the route to the destination and presents turn-by-turn instructions to the driver via text-to-speech, like existing navigation systems. In the background, RoadRunner also determines whether any regions it will traverse are congestion-controlled and attempts to obtain the corresponding tokens from the server via the cellular connection or from other vehicles via the V2V radio. The pseudocode for the overall logic of RoadRunner is shown in Algorithm 1.

RoadRunner leverages V2V communications to pass tokens directly between cars when possible, using the protocol described in Algorithm 2. Each vehicle broadcasts an *ANNOUNCE* message every 2 seconds, containing the vehicle’s ID, location, speed, bearing, and region IDs of tokens currently offered by the vehicle. *ANNOUNCE* messages are not rebroadcast or flooded because beyond 1-hop, the total latency for a token exchange exceeds V2Cloud latency: With a 40-millisecond latency for a 1-way V2V message, a 2-hop token exchange incurs 4 V2V messages with a total latency of 160 ms versus V2Cloud’s 140 ms latency.

```

if starting trip then
    determine route to destination;
    determine which tokens need to be obtained;
    request tokens from server over cellular connection;
    if not all necessary tokens are obtained then
        add unfulfilled requests to tokensWanted queue;
        retry periodically to server;
    end
    speak prompt to driver to begin driving;
end
if nearing region then
    if token for the upcoming region has not been obtained
    then
        make a final retry to the server for the token;
        if successful then
            continue with no changes to the route;
        else
            reroute to avoid region;
        end
    end
end
if entering a region then
    if the necessary token has been obtained then
        mark the token as in-use;
    else
        log infraction and enforce penalty;
        create short-lived PENALTY token for this
        region;
        (PENALTY token life is 10 minutes in our
        deployment);
    end
end
if exiting a region then
    if in-use token is a PENALTY token then
        destroy PENALTY token;
    else
        remove in-use designation from token;
        place token in a tokensOffered collection;
    end
end

```

Algorithm 1: Overall pseudocode

```

if received ANNOUNCE from another vehicle then
    foreach token in (tokensWanted) do
        if token is offered in ANNOUNCE message then
            send other vehicle TOKEN_REQUEST
            message for token;
        end
    end
end
if received TOKEN_REQUEST from another vehicle then
    if token requested is in our tokensOffered collection
    then
        remove the token from tokensOffered;
        send it in a TOKEN_SEND message to the other
        vehicle;
    else
        ignore this TOKEN_REQUEST;
    end
end
if received TOKEN_SEND from another vehicle then
    if we already possess a token for that region then
        place this extra token into the tokensOffered
        collection;
    else
        remove the now-fulfilled request from the retry
        queue tokensWanted;
        store the token for later use;
    end
end

```

Algorithm 2: Token exchange pseudocode.

Figure 1: Pseudocode for RoadRunner.

This *ANNOUNCE*, *TOKEN_REQUEST*, *TOKEN_SEND* hand-shake is necessary to ensure that each token is a singleton; if the tokens were grabbed directly from the *ANNOUNCE* message, multiple copies of the token may appear since multiple vehicles may hear the *ANNOUNCE* message. The *TOKEN_SEND* message includes the unique ID of the vehicle that is allowed to receive and use the token, so no duplicates occur among multiple vehicles. *TOKEN_REQUEST* messages arriving from multiple vehicles are processed in the order received, first-come first-served. Each *TOKEN_SEND* message is sent 3 times to minimize the possibility of token loss, and each *TOKEN_SEND* message includes a per-vehicle nonce so that extraneous receptions of the same *TOKEN_SEND* messages can be discarded. Tokens may still be lost over time, however, so we periodically expire all tokens and generate fresh tokens, in what we call **token roll-over** which occurs every hour: All the tokens in the system expire, and the server generates new tokens. Vehicles that possess an expired token can still use it, but will delete it upon usage (entering and traversing the corresponding region). This process has two purposes: 1) to ensure that any tokens lost during a V2V exchange are eventually re-injected into the system, and 2) to enable authorities to adjust the number of tokens while the system is operating, which would be difficult if it were unknown how many tokens remained in the system and might be reused indefinitely.

IMPLEMENTATION

We implemented RoadRunner as an Android application on Samsung Galaxy Note smartphones. The application consists of an Android Service (RoadRunnerService) that implements the main logic of RoadRunner and continuously runs in the background, and an Android Activity (Main-Activity) that shows the status of the application. RoadRunnerService and MainActivity run in the same thread. A separate thread manages the V2V communications interface (Wi-Fi or DSRC), running a busy-wait loop to receive packets from the network interface associated with the V2V radio. We were not concerned with energy consumption as the smartphone can be plugged into the vehicle's power source.

The smartphones communicate with the remote server over 4G LTE cellular data, which represents the state-of-the-art in mobile data access today. On the server, we implemented a Python application that services requests over TCP through a line-based protocol, allowing vehicles to make requests (a GET request) for and receive tokens from the server, and to send tokens back to the server (a PUT request). If there are no tokens available for a requested region, the server will respond with an error code (GET 500 FULL).

For V2V communications, the app can leverage either 802.11p DSRC or 802.11n adhoc Wi-Fi. We implemented support for both interfaces in our Android application, and include 802.11n in our evaluation to demonstrate that WiFi's range limitations render it inadequate for V2V communications.

To use 802.11p DSRC, we connect the Android smartphone to a Cohda Wireless MK2 WAVE-DSRC Radio [3] through the MK2's USB 2.0 OTG host interface. We enable USB tethering on the smartphone, which enumerates the phone as a generic USB CDC Ethernet interface on the MK2 host. FwdWsm, a software bridge application on the MK2, receives UDP packets from the phone on the ethernet interface, encapsulates them in WAVE Short Message (WSM) packets, and broadcasts them over the 802.11p wireless interface. FwdWsm also receives WSM packets, removes the WSM headers, and forwards the resulting UDP packets to the USB Ethernet interface. Thus, our Android app can communicate with the 802.11p radio by sending and receiving UDP packets. The MK2 radios utilize dual roof-mounted 5.9 Ghz antennas, and are powered by the 12V power supply from the vehicle's cigarette lighter port.

To use 802.11n adhoc Wi-Fi, we run Android 2.3 on the Galaxy Note smartphones as we require support for wireless extensions (WEXT), available only in the Android 2.3 drivers for the Broadcom BCM4330 chipset. We cross-compiled *iwconfig* to configure the cards into adhoc mode at the lowest bitrate supported (1Mbps) with power management disabled. Each smartphone is configured with a unique static IPv4 address, and all V2V communications happens over UDP broadcast.

Practical considerations

Electronic tolling/road-pricing. While RoadRunner is implemented and deployed as an area-space rationing system, it can be readily modified to support conventional road pricing. A time-based road pricing scheme, such as that used in Singapore's ERP [31], where a different rate is charged at different times, can be straightforwardly implemented with RoadRunner by having the server attach prices to tokens at the start of a time interval when first distributing, and having tokens expire at the end of a time interval. Every time a vehicle receives a token, it deducts the corresponding value from the vehicle's account. Thus, every time the tokens expire, a new batch of tokens with new prices can be generated on the server.

Cars already in the region can contact the server a short time (randomized to spread out V2Cloud

load) before the deadline and trade in their soon-to-expire tokens for fresh ones, while any tokens not traded in would be considered lost and freshly regenerated by the server. While this would require every car holding tokens to contact the server, RoadRunner would still have most of its beneficial effects on the cellular network as token exchanges that occur outside of token renewals can still be offloaded to V2V. Transportation authorities can tune how long-lived tokens are to trade off between more frequent V2V offload and more frequent token renewals and pricing updates.

The payment account can be implemented via a prepaid smartcard inserted into an interface to the smartphone, as is currently done with a specialized in-vehicle unit in Singapore’s ERP [31] system, or it can be electronically managed via online accounting systems like PayPal.

Vehicles origin or destination within a region. If a vehicle ends a trip or parks in a region that it currently has a token for, it can return the token to the cloud server. If a vehicle begins a trip in a congestion-controlled region, transportation authorities can choose how to enforce policy: 1. vehicles originating in a region may not begin moving (detected from GPS speed) until RoadRunner acquires a token for the region, or 2. vehicles originating in a region do not count towards that region’s quota.

Failure of cloud server and/or cellular network. In the event of a cloud server or cellular network failure, the system can still operate purely over V2V, as vehicles will indefinitely offer tokens over V2V if they cannot return them to the cloud server, and other vehicles can still request those tokens over V2V communications. Since vehicles are no longer able to exchange tokens via V2Cloud communications (GET from and PUT to cloud), there may be more unnecessary PENALTY infractions as some vehicles may never come within V2V range of desired tokens. Vehicles which end their trips out of V2V range will cause the tokens they are holding to be lost, which will cause a decrease in the total number of tokens in the system over time. Thus RoadRunner can survive V2Cloud communications outages, but only for a limited duration.

DEPLOYMENT OF ROADRUNNER PROTOTYPE

We evaluated RoadRunner by comparing the performance of RoadRunner with V2V communications and token exchanges enabled to a Cloud-only baseline. The Cloud-only variant communicates solely with a remote server via an LTE cellular connection, and the V2V-enabled variant additionally communicates with other vehicles over 802.11p DSRC. To demonstrate the inadequacy of adhoc Wi-Fi for V2V communications, we also evaluated a variant that used adhoc WiFi instead of 802.11p DSRC. The server portion was implemented as a Python application that serviced requests over HTTP, and was located in the same geographic region as the phones to minimize backbone Internet latency.

Our deployment took place in eastern Cambridge, MA, USA (Figure 3), a triangular region with a base width of 775 meters and a height of 315 meters. We split each road into regions between intersections which served as the buffer zones, resulting in a total of 11 regions on 2570 meters of road, with 4 controlled regions having a bounded number of tokens available for each, and 7 unrestricted regions that did not require tokens to traverse (Figure 4). This resulted in a fine-grained congestion control scenario of 4 controlled regions with a total distance of 900 meters, whereas Singapore’s Orchard Road ERP zone is 1 controlled region of 2200 meters, an order of magnitude coarser.

Ten vehicles participated in our experiment, driving along a default loop through Mass. Ave, Main St and Vassar St, with half of the vehicles going clockwise, and the other half going anti-clockwise. The RoadRunner app provided voice-over instructions to drivers to divert to Windsor St or Albany St depending on the success/failure in obtaining the necessary tokens. Vehicles circulated among the regions for 20 minutes beforehand to reach a random steady-state distribution of vehicles



Figure 2: Setup in each vehicle.

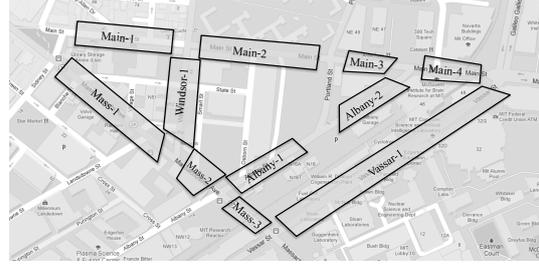


Figure 3: Deployment map with regions.

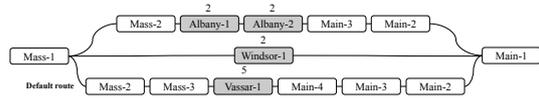


Figure 4: Routes in the deployment. (Controlled regions have capacity shown.)

over the deployment area. Each vehicle had two smartphones mounted on the windshield, one connected to a DSRC radio and the other utilizing its internal WiFi radio as in Figure 2.

We selected token quotas for each region to ensure that at least some of the ten vehicles had to divert onto alternative routes because they were unable to obtain a token, and to ensure that some vehicles would be unable to obtain tokens for any route and would incur a PENALTY reservation. Due to RoadRunner’s design, if a vehicle is in a congestion-controlled region or in an adjacent region, it will either have a token in-use or have an outstanding token request, except in the single intersection between Main-2 and Main-3 which is not adjacent to any congestion-controlled regions. Thus, our selection of token quotas and controlled regions in Figure 3 resulted in all the major scenarios: obtaining all necessary tokens for the default route, not obtaining tokens for the default route but obtaining necessary tokens for alternative route(s) alternative routes, and not obtaining necessary tokens for any route and thus incurring PENALTY reservations.

For the V2V-enabled variants, we tested the two possible operation paradigms of RoadRunner: 1) an on-demand navigation and routing system that requests tokens just-in-time for the next region, and 2) a pre-reserve system that requests all necessary tokens at the beginning of each trip iteration. For the Cloud-only variation, we did not test on-demand vs pre-reserve because they are effectively the same: all requests end up going through the remote server anyway, and unused tokens do not remain on the vehicles for V2V exchanges. Only the cloud server has any available tokens, so it does not matter whether vehicles check with the cloud at the beginning of a trip or on-demand.

We ran two ten-minute trials each of V2V on-demand over WiFi, V2V on-demand over DSRC, V2V pre-reserve over WiFi, V2V pre-reserve over DSRC, and Cloud-only.

On-demand requests versus pre-reserve requests. RoadRunner can operate in two system-wide modes for V2Cloud communications: Pre-reserve and On-demand. In both modes, vehicles watch for and attempt to obtain desired tokens over V2V communications at all times. These two modes represent a trade-off between providing more certainty about the route a driver will take at the beginning of a trip vs. improving token utilization, which directly impacts road usage.

Pre-reserve requests mode allows RoadRunner to make token requests for all the tokens it needs for a route at the beginning of a trip over V2Cloud. The vehicle periodically reattempts any remaining unfulfilled token requests, too, increasing the frequency of V2Cloud token requests in the system. Furthermore, vehicles hold tokens for a longer period of time without actively using them, decreasing token utilization rates. Pre-reserve requests may provide a better user experience, however, since the system can show the driver a complete, preferable route at the beginning of

the trip if available, rather than as a reroute while the user is already driving.

On-demand requests mode defers the V2Cloud token request for each region until the vehicle arrives at the intersection immediately before entering the region, reducing the frequency of V2Cloud communications and reducing the time that tokens may remain unused on a vehicle before it has reached the region. This may provide a poorer user experience, however: more preferable routes may become available only during the drive, requiring reroutes and imposing uncertainty.

Server retry timeout. Unfulfilled token requests are periodically retried to the server, determined empirically: 2 seconds for the cloud-only variant because tokens can only come from the server, 10 seconds for the V2V-enabled on-demand variant because tokens may come from other vehicles, and 30 seconds for the V2V-enabled pre-reserve variant because tokens are even less likely to be on the server since tokens are requested at the beginning of a trip.

Token PUT timeout. When a vehicle exits a region, it waits before returning the token to the server. A longer timeout increases the window for a V2V token exchange, but a shorter timeout decreases the duration a token may sit unused on the vehicle. We determined timeouts empirically: 10 seconds for the V2V-enabled on-demand variant because there are fewer unfulfilled requests in the system overall as requests are made on-demand when nearing a region, and 60 seconds for the V2V-enabled pre-reserve variant because there are more outstanding unfulfilled requests in the system overall as tokens are requested at the beginning of the trip.

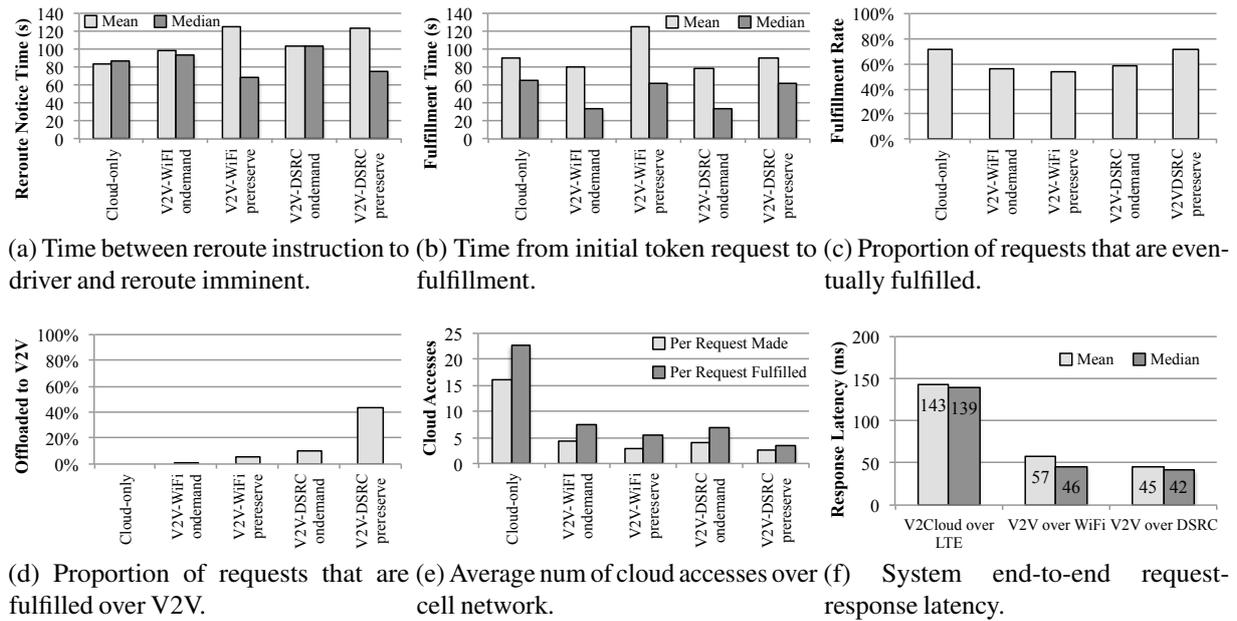


Figure 5: RoadRunner deployment measurements

Request fulfillment offload

Figure 5d shows the proportion of all fulfilled token requests that are fulfilled over V2V (in the Cloud-only variant, all requests are fulfilled over V2C). V2V-WiFi is not able to offload many token exchanges, due to the limited range of Wi-Fi: in the deployment, only 5 token exchanges occur over Wi-Fi at a mean distance of 29.2 meters, while 47 token exchanges occur over DSRC under the same conditions, at a mean distance of 175.7 meters. V2V-DSRC is able to offload a significant portion of token exchanges, up to 43%. The pre-reserve variants offload more than the

on-demand ones as requests for each region are made at the beginning of the trip rather than just before the region, giving vehicles more time to encounter a token offered over V2V.

Request fulfillment time

All token requests are timestamped when the request is created and when the request is finally fulfilled, whether by the centralized server (V2C) or by another vehicle (V2V). If the quota for a region is too low, fulfillment times may be unnecessarily long, and result in sub-optimal road throughput as cars are throttled waiting for tokens even when there is spare road capacity. If the token quota is too high, fulfillment times will be low, but roads may become congested. We graph the fulfillment times for the variants of RoadRunner in Figure 5b.

With good V2V communications, the RoadRunner distributed token reservation protocol is able to match the request fulfillment times of the Cloud-only baseline. Average fulfillment times for both DSRC and WiFi variants of RoadRunner are similar to the Cloud-only baseline, with the exception of the V2V-WiFi pre-reserve variant, which obtained only 5 tokens over V2V out of 73 total tokens obtained (a 6.8% ratio) due to the limited range of WiFi. In contrast, V2V-DSRC pre-reserve obtained 37 tokens over V2V out of 86 total, or 43% of tokens. DSRC's improved range allowed it to more often bypass the wait for a token to appear on the cloud, reducing the fulfillment time. On-demand variants have much lower median fulfillment times than cloud-only and pre-reserve variants because requests for a region are made just-in-time in the prior region.

Request fulfillment rate

The fulfillment rate is the proportion of token requests that eventually do acquire a token. The fulfillment rate is useful for understanding the effects of the more unreliable vehicle-to-vehicle communications and token exchange protocol vs. the reliable cellular and cloud server connection (available 91% of the time in our deployment). (Note that a fulfillment rate of 100% is undesirable since it is the same as no congestion control: every vehicle can enter a controlled region.) We use the V2C fulfillment rate as a baseline, and expect that V2V RoadRunner should result in similar fulfillment rates. Any significant deviations would imply that the V2V variants are negatively impacting the ability of vehicles to obtain tokens, beyond the effects of traffic congestion.

We show the fulfillment rates for the variations of RoadRunner in Figure 5c. DSRC RoadRunner show similar fulfillment rates to the Cloud-Only baseline, implying that when using DSRC for V2V communications, the distributed road reservation protocol of RoadRunner successfully fulfills token requests just as well as a Cloud-only implementation.

The WiFi variants show poorer fulfillment rate, indicating that the distributed road reservation protocol is negatively impacting the ability of vehicles to obtain tokens, due to WiFi not having sufficient range to meet other vehicles with tokens. Pre-reserve DSRC Roadrunner has better fulfillment rates than on-demand DSRC Roadrunner as token requests are created at the beginning of a trip rather than on-demand, giving the vehicle more time to encounter a nearby vehicle offering that token: indeed, DSRC pre-reserve was able to fulfill request over V2V more frequently (43.0% of all requests vs. 10.6% for DSRC on-demand).

Reroute notice time

The reroute notice time is the time from when the route changes (a reroute) due to a token being newly acquired, to when the driver turns onto the new route. Reroutes occur if a more preferable route becomes available; when this happens, we automatically update the navigation route to the most preferable, present updated turn-by-turn voice navigation directions to the driver, and display

tokens in possession on the screen. If the driver takes a different route or is unable to turn onto the new route in time, RoadRunner offers the tokens over V2V or returns them to the server.

In our deployment, the route passing through Windsor-1 is the shortest and most preferable, the route passing through Albany-1 and Albany-2 is the next most preferable, and the route through Vassar-1 is the least preferred. The Vassar-1 route is the default route presented to the driver, and if no tokens are available, the driver will incur a *PENALTY*.

The reroute notice times for the variants of RoadRunner are shown in Figure 5a. In all but one case in the Cloud-only variant, drivers had at least 50 seconds to turn onto the route.

The on-demand V2V variants of RoadRunner outperformed the Cloud-only baseline in reroute time provided to the driver. The pre-reserve V2V variants had a bimodal distribution: when vehicles are able to pre-reserve tokens in advance at the beginning of the trip, this counts as a reroute away from the default, longest route and thus those lucky drivers are afforded a large amount of time to take the new route. For drivers who did not get those tokens, however, they often get tokens just-in-time as the previous group of lucky drivers finish using their tokens and make them available to the latter group.

System responsiveness

We characterized the Vehicle-to-Vehicle (V2V) and Vehicle-to-Cloud (V2C) interactions in our deployment for an apples-to-apples comparison. All token exchanges are timestamped on the phones from request sent to response received to obtain end-to-end system latencies. We compare the latencies for interactions occurring Vehicle-to-Cloud over 4G LTE (V2C), Vehicle-to-Vehicle over adhoc Wi-Fi (V2V-WiFi), and Vehicle-to-Vehicle over DSRC (V2V-DSRC).

V2V latencies, shown in Figure 5f, are significantly lower than V2C latencies, with interactions over WiFi showing 61.2% reduction in mean latency and 22.5% reduction in median latency, and DSRC showing a 79.9% reduction in mean latency and 62% reduction in median latency. V2C latencies have a much higher mean than median due to a long-tail distribution in which some cellular accesses taking a disproportionately long time to complete. These findings are consistent with prior characterization studies [18, 20, 29].

DSRC latencies are not as low as the 100 microseconds delay requirement for safety applications or previously measured DSRC latencies [36], as we have additional delays incurred from the use of the FwdWsm software bridge, the USB Ethernet interface to the phones, the Android stack and Dalvik VM that Android apps run within, and the RoadRunner application overhead. Congestion control is not a safety application, however, and DSRC RoadRunner already shows significant improvements over the conventional client-server implementations of prior infrastructure-less electronic tolling systems [24] [21] [33] that rely solely on cellular.

Cloud access offload

For each of the RoadRunner variants, we measure the ability of the system to reduce the load on the cellular data network. For each variant, we divide the total number of requests made to the cloud server over the LTE connection by the number of token requests successfully fulfilled. Figure 5e shows that all V2V variants of RoadRunner are able to reduce cloud accesses per token significantly compared to the Cloud-only variant with reductions ranging from 66.3% to 84.3%.

These results demonstrate the benefits of leveraging V2V communications to exchange tokens over a conventional client-server implementation. We achieved reductions up to 84% in cloud accesses incurred per request, and latency reductions up to 80%. The RoadRunner distributed

token protocol running over DSRC matches the fulfillment rate of a Cloud-only baseline and does not significantly increase unnecessary penalties on controlled regions.

Due to our limited deployment size of 10 vehicles, it is difficult to measure RoadRunner’s effectiveness in mitigating traffic congestion at realistic scale: the token quotas are so low (only 2-5) that infractions are enforced when the region is not even close to capacity because some tokens have been reserved by cars not yet in the region. Instead, we rely on our following simulation studies (Section 6) to demonstrate the enforcement and congestion control effectiveness of RoadRunner.

LARGE-SCALE SIMULATION

While our deployments provided us with measurements of RoadRunner’s application and network performance, it is also critical to evaluate RoadRunner’s effect as a transportation policy vs. existing road pricing schemes. Policy-wise, a road-space rationing policy like RoadRunner allows drivers to enter road segments free-of-charge when the congestion level is below the quota, making it more palatable to drivers. Here, we evaluate RoadRunner’s effectiveness on transportation policy metrics like travel speed and road capacity/throughput vs. existing Electronic Road Pricing (ERP) in Singapore. We used the SimMobility short-term simulator (SimMobilityST) [10], an agent-based, multimodal microscopic simulator where drivers, pedestrians and passengers are modeled as agents whose behavior and movement are captured at a very fine resolution of milliseconds. SimMobilityST models detailed human behavior, including drivers changing lanes, accelerating/braking, choosing routes, and how pedestrians walk, how people board buses, etc.

We recreated the movement of vehicles under ERP and simulated individual driver behavior, vehicle movement, the RoadRunner app, and communications latency and range within SimMobilityST, for an existing road pricing region in Singapore, with realistic vehicle traffic generated from actual loop detector information, at 10-millisecond resolution (necessary to model network communications latencies of 50-200 ms).

We simulated the RoadRunner on-demand app with a congestion control policy providing various numbers of road reservation tokens on Orchard Road. We did not simulate RoadRunner pre-reserve as the loop count data does not provide true trip origins outside the Orchard Road area road network, which are necessary to simulate making pre-reserve requests at the beginning of a trip.

Traffic movement modeling. We start with loop detector counts from the Orchard Road ERP region in Singapore, collected over a 24 hour period beginning Thursday, August 5, 2010. Vehicle counts are available for eight intersections on the Orchard Road road pricing region (Figure 6). For each intersection, we manually annotated which detectors counted vehicles turning onto (entering) the region, which detectors counted vehicles turning out of (exiting) the region, and which detectors counted vehicles continuing to travel inside the region.

The raw loop count data provides the number of vehicles crossing through the intersection of each lane every 5 minutes, detected by loop detectors embedded under each lane. To simulate the movement of individual vehicles within the region, we create Origin-Destination (OD) pairs that represent vehicle trips. We assume a Poisson process for vehicle detections, and thus distribute detections randomly across the 5 minute time interval into 10 millisecond bins.

For each of the roads crossing Orchard Road, we select an Origin point on the crossroad 250 meters before the intersection, for each possible direction that a vehicle can enter from. Similarly, we select a Destination 250 meters after the intersection for each possible exit direction. Each loop detector is paired with the corresponding Origin and Destination point.

For each detected exit, we correlate it to a Destination point based on which intersection and lane

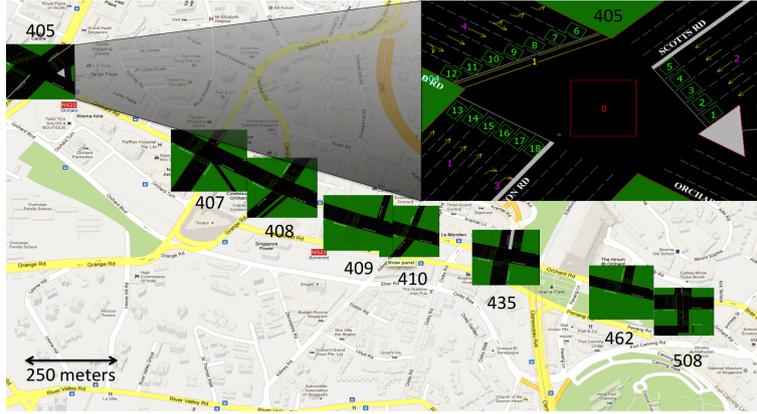


Figure 6: Map of Orchard Road and intersections for simulation.

the loop detector is in, and pair it with one of the valid Origins from which it could have come from, sampled from the distribution of entrance loop count detections in the same time-step. We assume that the distribution of cars across the entry points varies slowly relative to the travel time of a single vehicle within the 2.2 km long road. This resulted in 74,904 Origin-Destination pairs representing trips through Orchard Road.

Simulation setup and parameters

We simulated the three variants of RoadRunner On-demand (Cloud-only, V2V-Enabled with DSRC, V2V-Enabled with WiFi) and compared it to an unmodified simulation of the baseline ERP policy in which all the Origin-Destination pairs travel through Orchard Road.

With the RoadRunner variants, if RoadRunner cannot obtain a token, it finds an alternate route that avoids Orchard Road and reaches the Destination point via other roads near Orchard Road and reroutes the driver. If it cannot find an alternate route, the driver continues onto Orchard Road but incurs a penalty reservation, similarly to the real deployment.

We simulate a V2V communications range and latency of 175.7 meters and 45 milliseconds for DSRC and 29.2 meters and 57 milliseconds for WiFi, with a 100% message reception rate, based on the average token exchange distances and latencies in our real deployment (Section 5). We also simulate a cloud server that exchanges tokens with the vehicles over a V2Cloud cellular connection, with 143 ms latency and 100% availability. (The cellular data connection was available 91% of the time in our deployment, and 100% with retries).

We initialize the simulation with no vehicles within the region when the simulation begins at 12:00am. Over the course of the simulation, for the baseline ERP policy, this builds up to a maximum of 139 vehicles on Orchard Road at 11:37am.

At each time-step, SimMobility simulates driver behavior (navigation, lane changes, maintaining following distances, etc.) and the RoadRunner app, which operates in the same manner as the RoadRunner app in the deployment, with a few differences:

Localization. SimMobility provides the vehicle location directly to the app.

On-demand V2Cloud requests. As the vehicle travels along a crossroad, RoadRunner requests a token on-demand when the vehicle is 50-70 meters away from the intersection with Orchard Road (using a 100x100 meter bounding box)

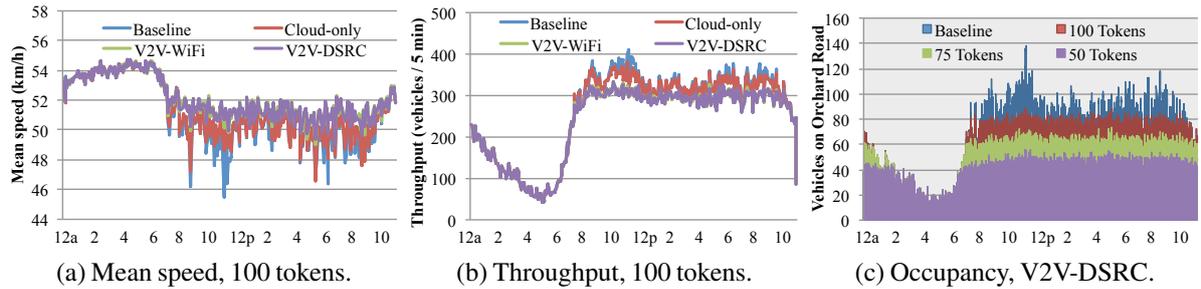


Figure 7: RoadRunner Orchard Road simulation measurements

Simulation results

Quota enforcement: Figure 7c compares the vehicular occupancy of Orchard Road for the Cloud-only, V2V-DSRC, and V2V-WiFi variants when 100 tokens are allocated. RoadRunner successfully reduces the number of vehicles in the region according to the number of tokens, even when some drivers cannot find an alternate route and thus must traverse Orchard Road anyway.

Travel speed and throughput: Figure 7a compares the average speed of vehicles with different token allocations for the V2V-DSRC variant. We sample the speed of every vehicle on Orchard Road once per second and aggregate the samples into 5-minute bins. RoadRunner improves the travel speed of vehicles at times of congestion compared to the ERP baseline. It significantly eliminates many peak congestion periods and improves minimum travel speeds: the slowest average travel speed across the 24-hour period improves 7.7% from 45.5 km/h in the ERP baseline to 49.0 km/h with V2V-DSRC. Overall vehicle throughput is reduced during peak congestion in our evaluation: some tokens are not circulated immediately to vehicles demanding them, and in other cases, improving travel speed necessarily reduces throughput. The number of tokens can be adjusted by transportation policy planners to trade-off between vehicle speed and total throughput.

Cellular data access reduction: In the simulations with 100 tokens allocated, the Cloud-only baseline recorded 73,693 cellular network accesses to GET or PUT tokens. The V2V-WiFi variant reduced the number of these cellular accesses by 14.5%, and the V2V-DSRC variant reduced cellular accesses by 24.8%. This reduction is lower than in the deployments because there is only one region, so no additional token exchanges occur between vehicles for different regions.

RELATED WORK

RoadRunner is related to several previous systems in infrastructure-less congestion control, but differs by leveraging direct V2V communications instead of relying purely on a centralized client-server design using the cellular data connection. It also realizes congestion control with road-space rationing rather than conventional road pricing, a policy that we believe is much more acceptable to the general public for a pervasive, city-wide implementation. Lu et al [24] and Lee et al [21] demonstrate GPS-based tolling systems with tolling information reported to a server through a GPRS or 3G cellular connection. Srinivasan et al [33] present a map matching and development platform for infrastructure-less electronic road pricing systems that runs on mobile devices, which can be applied to RoadRunner for more accurate localization.

RoadRunner is not a traditional vehicular network as it combines a reliable cellular connection and restricts vehicular routing to a single hop to keep response times low, but the following systems provide valuable insights on message routing, vehicular positioning, and security. Leontiadis et al [22] present a geographic routing protocol for vehicular networks and simulate using vehicle traces.

Wu et al's MDDV [35] is an algorithm for data dissemination over V2V that combines opportunistic, trajectory based, and geographical forwarding, applicable to keeping tokens geographically near their regions. MaxProp [13] routes message between peers without knowing the state of a partitioned disruption-tolerant network or the meeting locations. Wisitpongphan et al [34] show that conventional routing techniques such as AODV or DSR do not work for sparse vehicular adhoc networks, such as on a RoadRunner controlled region during times of low traffic. Boukerche et al [12] examine the suitability of data fusion techniques to provide robust localization for vehicular networks, which could help improve our controlled region granularity. Parno et al [26], Raya et al [28], and Lin et al [23] contribute protocols, discussion, and designs on securing vehicular networks, critical to ensuring malicious users do not defraud or disable RoadRunner.

CONCLUSION

In this paper, we demonstrated and realized a congestion control system with a distributed, infrastructure-less vehicular app that combines ubiquitous smartphones with vehicle-to-vehicle communications. Our experimental deployments show sufficient enforcement accuracy, faster response times, and effective offload of cellular network load when compared to a traditional client-server implementation. Our simulation results show that RoadRunner is more effective as a transportation policy than existing electronic road pricing. With device-to-device networking improving rapidly and becoming widespread in modern mobile devices, infrastructure-less smartphone apps such as RoadRunner can enable low cost and pervasive intelligent transportation services.

ACKNOWLEDGEMENTS

We thank the many volunteer drivers in our deployments, and Kakali Basak, Seth Hetu, Xu Yan, and Vahid Hamishagi for help with the SimMobilityST integration. This research was supported by the Singapore National Research Foundation through the Singapore-MIT Alliance for Research and Technology (SMART)'s Future Urban Mobility (FM) research program, and by the United States Department of Defense NDSEG fellowship.

REFERENCES

- [1] Applications of road space rationing. en.wikipedia.org/wiki/Road_space_rationing#Applications_of_road_space_rationing.
- [2] List of Electronic Toll Collections Systems. en.wikipedia.org/wiki/List_of_electronic_toll_collection_systems.
- [3] MK2 WAVE-DSRC Radio. cohdawireless.com/product/mk2.html.
- [4] Mobile Broadband Capacity Constraints And the Need for Optimization. rysavey.com/Articles/2010_02_Rysavy_Mobile_Broadband_Capacity_Constraints.pdf.
- [5] Verizon Killing Unlimited Data Plans Soon. huffingtonpost.com/2011/06/21/verizon-unlimited-data-plan_n_881091.html.
- [6] Netizens in singapore spend more than a day a week on internet, 2012.
- [7] Smartphones account for half of all mobile phones, 2012.
- [8] Cisco, NXP Invest in Cohda Wireless to Enable the Connected Car. newsroom.cisco.com/press-release-content?articleId=1119826, 2013.
- [9] M. Almendoar. Call for new erp proposals. *Straight Times*, June, 2010.
- [10] K. Basak, S. Hetu, et al. Modeling reaction time within a traffic simulation model. In *IEEE Intelligent Transportation Systems*. 2013.
- [11] R. Bose, J. Brakensiek, et al. Morphing smartphones into automotive application platforms. *IEEE Computer*, 44(5), May 2011.
- [12] A. Boukerche, H. Oliveira, et al. Vehicular ad hoc networks: A new challenge for localization-based systems. *Comp. Comm.*, 31(12), 2008.

- [13] J. Burgess, B. Gallagher, et al. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *IEEE INFOCOM*. 2006.
- [14] C. Daganzo. A pareto optimum congestion reduction scheme. *Transportation Research Part B: Methodological*, 29(2), 1995.
- [15] J. Dunn. Cars Get Connected: The Best Car Tech Trends Today. technologyguide.com/default.asp?newsID=5255, Dec. 2012.
- [16] J. FINN. The rebirth of toll: Etc makes its american comeback. *Tolltrans. Traffic Tech. Intl. Supplement*, 2004.
- [17] M. Goh. Congestion management and electronic road pricing in singapore. *Journal of Transport Geography*, 10(1), 2002.
- [18] J. Huang et al. A close examination of performance and power characteristics of 4g lte networks. In *ACM MobiSys*. 2012.
- [19] C. Keong. Road pricing: Singapores experience. In *IMPRINT-EUROPE Thematic Network, Implementing Reform on Transport Pricing: Constraints and Solutions: Learning from Best Practice, Brussels*. 2002.
- [20] E. Koukoumidis et al. Pocket cloudlets. *ACM SIGARCH Computer Architecture News*, 39(1), 2011.
- [21] W.-H. Lee, B.-S. Jeng, et al. Electronic toll collection based on vehicle-positioning system techniques. In *IEEE ICNSC*. 2004.
- [22] I. Leontiadis and C. Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In *IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks*. 2007.
- [23] X. Lin et al. Security in vehicular ad hoc networks. *IEEE Commun. Mag.*, 46(4), 2008.
- [24] S. Lu, T. He, et al. Electronic toll collection system based on global positioning system technology. In *IEEE Challenges in Environmental Science and Computer Engineering (CESCE)*, vol. 2. Mar. 2010.
- [25] K. Nakamura and K. Kockelman. Congestion pricing and roadspace rationing: an application to the san francisco bay bridge corridor. *Transportation Research Part A: Policy and Practice*, 36(5), 2002.
- [26] B. Parno and A. Perrig. Challenges in securing vehicular networks. In *HotNets-IV*. 2005.
- [27] S. Phang and R. Toh. From manual to electronic road congestion pricing: The singapore experience and experiment. *Transportation Research Part E: Logistics and Transportation Review*, 33(2), 1997.
- [28] M. Raya and J. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1), 2007.
- [29] M. Satyanarayanan, P. Bahl, et al. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Comput.*, 8(4), 2009.
- [30] D. Schrank and T. Lomax. *The 2007 urban mobility report*. Texas Transportation Institute, Texas A & M University, 2007.
- [31] F. T. Seik. An advanced demand management instrument in urban transport: Electronic road pricing in singapore. *Cities*, 17(1), 2000.
- [32] J. Sonnenberg. Service and user interface transfer from nomadic devices to car infotainment systems. In *AutomotiveUI*. 2010.
- [33] D. Srinivasan, R. Cheu, et al. Development of an improved erp system using gps and ai techniques. In *Intelligent Transportation Systems*. 2003.
- [34] N. Wisitpongphan, F. Bai, et al. Routing in sparse vehicular ad hoc wireless networks. *IEEE J. Sel. Areas Commun.*, 25(8), 2007.
- [35] H. Wu, R. Fujimoto, et al. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In *ACM VANET*. 2004.
- [36] J. Yin, T. ElBatt, et al. Performance evaluation of safety applications over dsrc vehicular ad hoc networks. In *ACM VANET*. 2004.
- [37] Y. Zhou, Y. Wu, et al. The impact of transportation control measures on emission reductions during the 2008 olympic games in beijing, china. *Atmospheric Environment*, 44(3), 2010. ISSN 1352-2310.