www.zte.com.cn/magazine/English

ISSN 1673-5188
CODEN ZCTOAK

ZTE COMMUNICATIONS

An International ICT R&D Journal Sponsored by ZTE Corporation

March 2014, Vol.12 No.1

SPECIAL TOPIC: Vehicular Networks

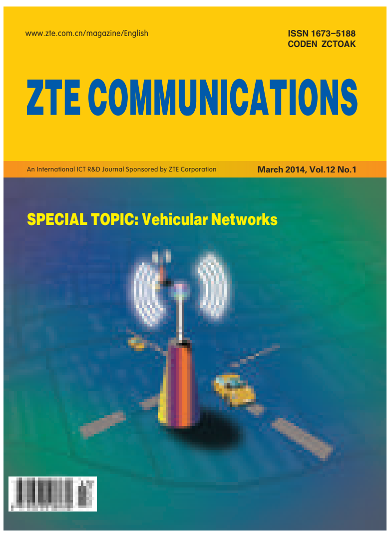
ZTE中兴

ZTE COMMUNICATIONS

VOLUME 12 NUMBER 1 MARCH 2014

ZTE Communications Editorial Board

CONTENTS



Special Topic

Vehicular Networks

CONTENTS

ZTE COMMUNICATIONS
Vol. 12 No.1 (Issue 41)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
ZTE Corporation and Anhui Science and Technology Information Research Institute

Editorial Correspondence:
Add: 12F Kaixuan Building,
329 Jinzhai Road,
Hefei 230061, P. R. China
Tel: +86-551-65533356
Fax: +86-551-65850139
Email: magazine@zte.com.cn

Published and Circulated (Home and Abroad) by:
Editorial Office of *ZTE Communications*

Printed by:
Hefei Zhongjian Color Printing Company

Publication Date:
March 25, 2014

Publication Licenses:
ISSN 1673-5188
CN 34-1294/ TN

Advertising License:
皖合工商广字0058号

Annual Subscription:
RMB 80

Research Papers

Roundup

Vehicular Networks

Jiannong Cao

Jiannong Cao is currently a chair professor and head of the Department of Computing at Hong Kong Polytechnic University. He received his BSc degree in computer science from Nanjing University, China, in 1982. He received his MSc and PhD degrees in computer science from Washington State University in 1986 and 1990. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He has co-authored four books, co-edited nine books, and published more than 300 technical papers in major international journals and conference proceedings. He has directed and participated in numerous research and development projects and, as a principal investigator, obtained more than HK$25 million in grants. He is the chair of Technical Committee on Distributed Computing, IEEE Computer Society; a senior member of IEEE; a member of ACM; and a senior member of China Computer Federation. He has been an associate editor and member of editorial boards of many international journals. He has been the chair and a member of organizing/program committees for many international conferences.

Vehicular networks have attracted a lot of attention in recent years. Many projects have been initiated by government, industry, and academia to improve driving safety, provide travel assistance, improve traffic flow, and decrease fuel consumption. These projects exploit vehicular communications and networking technologies, generally referred to as vehicular ad-hoc networks (VANETs) or more generally, vehicular networks. VANET includes vehicle-to-infrastructure and vehicle-to-vehicle communications and can be based on short-range and medium-range communication as well as cellular systems. The development and deployment of vehicular networks is also considered one of the critical foundations of the intelligent transportation system industry.

The purpose of this special issue is to explore recent developments in VANETs and stimulate discussion on future research on large-scaled VANET applications. This special issue comprises six papers covering the most active areas of research in VANETs. Three papers are dedicated to network communication protocols. Respectively, these papers describe network coding, QoS in V2I communications, and data routing schemes. Another two papers focus on novel VANET applications and services. One of these papers proposes real-time routing and the other proposes network-enhanced GPS techniques. The last paper describes a practical prototype vehicle system comprising connected vehicles.

The first paper by Yuanguo Bi et al. describes the problem of satisfying the diverse QoS requirements of various vehicular applications and efficiently utilizing limited wireless channel resources. The authors propose a cross-layer rate-control scheme to solve this problem. By tracking throughput, aggregate arrival rate, and buffer size at the bottleneck RSU in a timely manner, source sending rates can be adjusted in order to avoid buffer overflow or underflow at the RSU. In addition, bandwidth can be efficiently allocated to different types of multimedia services by taking the bandwidth requirements of each service into account. Simulation results show that this solution improves system throughput and ensures wireless channel resources are efficiently used. This solution also satisfies the different QoS requirements of multimedia services in V2I networks.

The second paper "Advanced Leader Election for Virtual Traffic Lights" describes a novel vehicular networking application called Virtual Traffic Light (VTL). This solution replaces current infrastructure-based traffic lights. The authors present the results of an evaluation of an improved VTL system designed to manage arbitrary intersection geometries. The effect of VTL on traffic flow and overall network performance is determined for two scenarios: a realistic urban road network and a synthetic single intersection. The results show that VTL positively affects the driving experience of simulated vehicles.

The third paper "Trajectory-Based Data Forwarding Schemes for Vehicular Networks" describes two data-forwarding schemes based on vehicle trajectory in VANETs. Vehicle trajectory is a good asset when designing data-forwarding schemes for multihop V2I or I2V data delivery because it allows for either a better forwarding metric computation or better estimation of the location of the packet destination vehicle.

The paper "Unveiling the Challenges for Improving Data Availability in Vehicular Networks with Network Coding" describes a network-coding-based approach for improving data availability in VANETs. An empirical study with trace-driven simulations shows that performance can be significantly improved, but codes received by a retrieving node tend to be linearly correlated. This is a serious issue because linear correlation of codes degrades data retrieval. The authors show this by designing synthetic networks that have community structures with dramatically different features and conducting experiments on these networks. The authors also discuss opportunities for improving network-coding-based approach for data availability in vehicular network by developing community-aware techniques for codes distribution.

The paper "Networking-GPS: Cooperative Vehicle Localization Using Commodity GPS in Urban Area" describes a new cooperative vehicle localization algorithm called Networking-GPS. The objective of this algorithm is to mitigate the multipath effect, which can severely degrade the localization performance of GPS receivers. The correlation between the similarity of SNR values from different GPS satellites and relative distance among different GPS receivers is identified. An evaluation based on real GPS traces shows that Networking-GPS can improve accuracy in the face of the multipath effect.

Finally, the paper "Anatomy of Connected Cars" describes the Internet of Vehicles, which is poised to become a reality as connected vehicles are mandated by the US Department of Transportation. This paper identifies the main characteristics of IoV information flow and two roadblocks to IoV deployment. The authors argue exploitation of community WiFi connectivity and shifting from a node-centric paradigm to a content-centric paradigm can accelerate IoV deployment and reduce costs.

The papers in this special issue are a snapshot of current issues in the field of VANETs. Large-scale VANET application is still in its infancy, and much current work focuses on the development of applications. There is a gap between the deployment of VANET applications and the existing vehicular communication protocols. The papers in this special issue show how that gap is closing.

We are grateful to all authors who submitted their papers for publication in this special issue. We thank Zongjian He, a PhD student at Hong Kong Polytechnic University for his great effort and enthusiastic support for this special issue.

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 001
http://www.cnki.net/kcms/detail/34.1294.TN.20140306.1003.001.html, published online March 6, 2014

End-to-End Rate Adaptation to Support Heterogeneous Services for Infrastructure-Based Vehicular Networks

Yuanguo Bi[1], Hangguan Shan[2], Xuemin (Sherman) Shen[3], and Hai Zhao[1]
(1. College of Information Science and Engineering, Northeastern University, Shenyang 110819, China;
2. Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China;
3. Department of Electrical and Computer Engineering, University of Waterloo, Waterloo N2l 3g1, Canada)

Abstract

Vehicle to Infrastructure (V2I) communications aim to provide mobile users on the road low-cost Internet and driver safety services. However, to meet Quality of Service (QoS) requirements of various applications and efficiently utilize limited wireless channel resources, the transport layer protocol has to perform effective rate control in low channel quality and frequent changing topology communication environment. In this paper, we propose a novel rate-control scheme in infrastructure based vehicular networks that avoids congestion and starvation and promotes fairness in end-to-end V2I communications. In vehicular networks, a bottleneck roadside unit (RSU) keeps track of its buffer size, aggregate incoming rate, and link throughput, and appropriately allocates bandwidth to traversing flows. With feedback information from the RSU, source nodes dynamically adjust their sending rates to avoid buffer overflow or starvation at the bottleneck RSU. Simulation results show that the proposed scheme can reduce not only packet losses owing to buffer overflow but also buffer starvation time, which improves the utilization efficiency of wireless channel resource.

Keywords

vehicle to infrastructure communications; roadside unit; congestion avoidance; starvation avoidance; transport layer

1 Introduction

Nowadays, mobile users are spending more and more time in their vehicles and are eager to access Internet on the road. These users need elastic best-effort services, such as web browsing, digital map downloading and e-mail, as well as inelastic real-time services, such as VoIP, video steaming, online gaming and traffic accident alarm, through fixed roadside infrastructure. In a V2I network, an roadside unit (RSU) is usually serves as a relay node that forward packets between wireline servers and mobile users within its communication range [1], [2] when mobile users establish end-to-end connections and request multimedia Internet services from wireline servers. However, limited and time-varying wireless bandwidth, which is induced by interference, packet collisions, channel fading and high mobility, usually leads to buffer overflow or starvation at an intermediate RSU if wireline source nodes are sluggish to adapt their sending rates to the throughput variation at the RSU. This greatly reduces the utilization efficiency of channel resources.

To efficiently utilize wireless resources in hybrid V2I networks, the transport layer has to have a flexible rate-adjustment mechanism to avoid buffer overflow or starvation at intermediate RSUs. However, there are several tough issues in designing an efficient rate control scheme for V2I networks. First, an RSU usually becomes a bottleneck when packets traverse from high-bandwidth wireline networks to low-bandwidth wireless networks. This leads to buffer overflow at the RSU. Second, within the limited coverage of an RSU, the number of active flows changes dynamically because of the high speeds of vehicles. When some vehicles move out of the coverage of an RSU, the buffer of the RSU may become starved if the vehicles' released channel resource cannot be distributed to other active flows. Third, an RSU may employ the automatic bit-rate selection mechanism to dynamically adjust its physical layer data rate according to the communication environment. This makes the media access control (MAC) layer throughput of the RSU quite unstable. If MAC layer throughput varies greatly, sluggish rate adjustments of wireline source nodes may lead to buffer overflow or starvation at the RSU. Finally, because the V2I network is dedicated to supporting heterogeneous services for mobile users, allocating bandwidth to traversing flows at the bottleneck RSU is difficult. The QoS requirements of both *elastic* and *inelastic* services need to be taken into account. In the complex V2I communication environment, wireline source nodes have difficulty choosing sending rates that avoid congestion and also effectively utilize the wireless channel resource at the RSU.

In this paper, we propose an efficient vehicular rate control (VRC) scheme for dynamically adjusting source sending rates and improving wireless resource utilization in V2I networks. The main contributions of VRC are: 1) It enables an RSU to keep track of its buffer size, packet-arrival rate, and MAC layer throughput in a timely manner in order to avoid buffer overflow or starvation in the dynamic communication environment. Rates are fed back so that source sending rates can be promptly adjusted. 2) VRC properly allocates bandwidth to each traversing flow according to the QoS requirements of both elastic and inelastic services. This enables efficient and fair sharing of the limited bandwidth at the bottleneck RSU.

In order to study the performance of our proposed VRC, we ran many simulations in the Network Simulator (NS-2). The result showed that that the proposed scheme improves the performance of different types of services.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 describes the system model and characterizes the QoS requirements of each type of multimedia service. Section 4 describes the details of the proposed VRC scheme. Section 5 presents numerical results. Section 6 concludes the paper.

2 Related Work

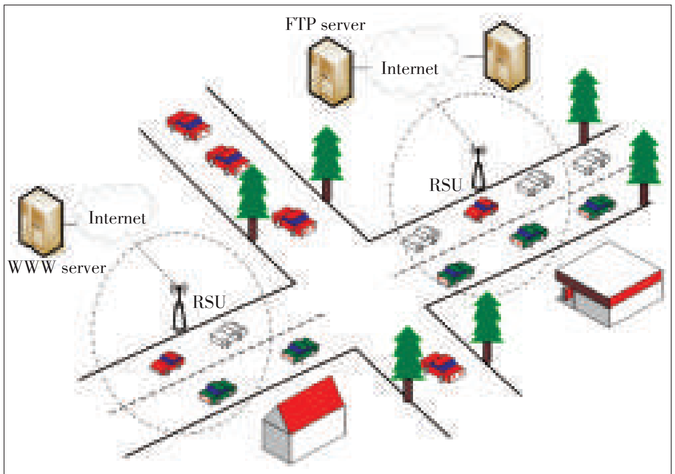
To control congestion, some Additive Increase Multiplicative Decrease (AIMD) algorithms [3], [4] take an end-to-end approach to adjusting source sending rates without the knowledge of the available MAC layer capacity. However, the long convergence time of AIMD-based schemes adversely affects the goodput of flows in a dynamic communication environment [5], e.g., the V2I communication scenario. To avoid long convergence time, some centralized bandwidth allocation schemes have been proposed to enable source nodes to quickly and accurately adapt their sending rates to time-varying link bandwidth. In XCP [6], an intermediate router calculates the available bandwidth using the medium capacity, previously allocated bandwidth, and persistent queue length. The bandwidth is equally allocated to traversing flows if the spare bandwidth is positive, but the bandwidth is distributed proportionally to the current flow rates when the spare bandwidth is negative. WXCP [7] considers MAC layer properties, such as the medium busy and idle times, and aims to estimate current link capacity more precisely. In [8], an improved algorithm is presented. This algorithm infers the available bandwidth by considering queue dynamics over variable-capacity networks. In [9], the Rate Control Protocol (RCP) emulates processor sharing to control congestion, and a router assigns a single rate to all flows that pass through it. However, [6]–[9] do not take into account channel utilization and packet loss, and describe a similar formula with constant parameters α and β for calculating the available bandwidth. This means that available bandwidth of the wireless link cannot be accurately determined [10], [11]. Although these centralized allocation schemes tend to achieve transport-layer fairness, the QoS requirements of different services are usually neglected, e.g., equally allocated bandwidth may not meet the QoS requirements of some inelastic services because such services usually cannot operate below the minimum required bandwidth.

There are several QoS-aware congestion control schemes in the literature. In [12], QoS utility functions are assumed to be continuously increasing over the available bandwidth. The authors propose a novel distributed flow control algorithm for service-differentiation networks. The algorithm has a new resource allocation criterion and achieves utility proportional fairness by paying special consideration to bandwidth sensitive real-time applications. In [13], the queue back-pressure-based stochastic optimization algorithm is extended to support both elastic and inelastic traffic in wireless sensor networks and to ensure fairness for non-concave utility-based inelastic traffic. This algorithm then implements a rate-control protocol for heterogeneous services in TinyOS-2.x operating system. In [14], the utility-based fairness is defined for time-constraint flows in vehicular networks. Both router-assisted and end-to-end congestion control schemes are based on the new concept of fairness. In [12]–[14], the sum of user utilities is maximized, and different utility functions are used for optimal bandwidth allocation. However, encoding all the desired effects, such as QoS requirements and fairness in the utility function, is nontrivial. To achieve the maximum number of user utilities in the dynamic V2I environment, a large amount of control overhead is induced by frequent bandwidth allocation and makes resource use much less efficient.

3 System Model

We consider a V2I communication network comprising wireline servers, RSUs, and moving vehicles[1] along a multilane road (**Fig. 1**). After successfully associating a nearby RSU, moving vehicles communicate with wireline servers through the associated RSU to obtain multimedia services, such as video streaming, file downloading, database access, and even safety alert. An intermediate RSU forwards data packets from wireline servers to moving vehicles within its fixed transmission range *R*, and usually tends to become the bottleneck because of its limited wireless bandwidth in V2I communications. Nevertheless, because an RSU can feasibly obtain knowledge of its buffer dynamics and wireless bandwidth variations, the proposed VRC scheme is implemented at the RSU to avoid congestion or starvation and allocate bandwidth. In stable and reliable high-bandwidth wireline networks, the source nodes dynamically adjust their sending rates according to rate feedback

[1] In order to simplify the expressions, we consider a vehicle as a communication entity instead of a fixed onboard communication unit or portable device (laptop, PDA, smart phone, etc.) in the following sections of the paper.



▲**Figure 1. The architecture of the V2I communication network.**

information from the RSU.

Heterogeneous services in V2I networks are divided into two major categories according to their bandwidth requirements: 1) elastic best-effort (EBE) services, i.e., traditional Internet services such as file downloading, web browsing, electronic mail, and remote terminal that all require acknowledgments (ACKs) to guarantee reliability but can tolerate rate variations; and 2) inelastic real-time services that usually do not need ACK feedbacks. Type-2 services are further divided into three categories:

- Hard real-time (HRT) services, such as road safety related applications, VoIP, and video conferencing. These have strict delay constraints, e.g., a delayed traffic accident alerts may lead to the loss of a traveler's life.
- Real-time rate-adaptive (RTRA) services such as IPTV and online streaming. These can tolerate some rate fluctuations in response to dynamic communication environment.
- Real-time delay-adaptive (RTDA) services. Some real-time audio and video services can tolerate occasional violations of latency requirements due to traffic congestion.

Generally, EBE services can adapt to bandwidth fluctuations; however, HRT services have stringent bandwidth requirements. RTRA and RTDA services are soft real-time (SRT) services. These usually have target sending rates but require minimum bandwidth guarantee because some audio and video coding schemes demand the lowest operating bit rate, below which the coder cannot function normally [15]. The goal of our work is to avoid buffer overflow or starvation and efficiently utilize wireless channel resource when heterogeneous services traverse through the bottleneck RSU.

4 Proposed Rate-Control Scheme

In this section, we discuss the design details of the proposed rate-control scheme. First, an RSU precisely keeps track of the aggregate arriving rate, link throughput, and its buffer size. Then, it feeds back the rate to adjust the source sending rates and avoid congestion or starvation or respond to buffer overflow and starvation. The end-to-end rate-control approach depends on source nodes that tentatively probe the bandwidth of the bottleneck link. However, our proposed VRC takes a centralized approach to bandwidth allocation at the intermediate RSU.

In V2I communications, the dynamic communication environment and fast-changing topology make the wireless bandwidth fluctuate frequently at the RSU. To reduce system overhead, an RSU does not need to take into account reactions to bandwidth variations caused by packet collisions, vehicle mobility, and medium contention. Instead, it keeps track of its buffer size and feeds back the rate to source nodes when its buffer is going to overflow or starve [16]. A tagged RSU calculates its latest throughput μ every period; that is,

$$\mu = \left(\sum_{i=1}^{n} x_i\right) \Big/ (T - t_s) \tag{1}$$

where n is the number of successfully delivered packets during T, x_i is the size of the ith successfully delivered data packet, and t_s is the cumulative starvation time of the buffer in period T. As well as periodically calculating the latest throughput, the RSU also keeps track of packets arriving at its buffer. The latest buffer arrival rate is given by λ, so that

$$\lambda = \left(\sum_{i=1}^{m} y_i\right) \Big/ T \tag{2}$$

where m is the number of arrived packets in period T, y_i is the size of the ith arriving packet during period T. With the parameters μ and λ, the RSU can avoid congestion and starvation and react to congestion and starvation.

4.1 Avoiding Congestion

When more contending vehicles move into the coverage area of the RSU, or if a hostile communication environment leads to decreased MAC layer throughput, the buffer of the RSU may start to build up; that is, $\lambda > \mu$. Therefore, if the latest aggregate incoming rate λ is larger than throughput μ, and

$$q + 2T_{f\max}(\lambda - \mu) \geqslant Q_{\max} \tag{3}$$

where $T_{f\max}$ is the maximum feedback delay from the RSU to a wireline server, q is its current buffer size, $Q_{\max}$ is the upper bound of the buffer size for avoiding buffer overflow, the RSU should reduce the aggregate incoming rate to its current throughput μ. In V2I networks with mixed services, different types of services have different requirements in terms of bandwidth, and bandwidth has to be properly allocated to each of these services. For services traversing the tagged bottleneck

RSU, we denote the set of EBE services S_e; we denote the set of HRT services S_h; and we denote the set of SRT (e.g., RTRA and RTDA) services S_s. To promote fairness among EBE services, we allocate the same bandwidth to each of the services and denote their feedback rates r_e. For SRT services, we allocate the feedback rate $r_s[i]$ to service i according to the service's target rate $R_{st}[i]$ and minimum required rate $R_{sm}[i]$. HRT services usually operate at a constant bit rate (CBR) regardless of traffic conditions at the bottleneck link. The constant sending rate of an HRT service I is given by $R_h[i]$. We divide link throughput μ into two hard allocated bandwidth (B_h) and soft allocated bandwidth (B_s).

Hard allocated bandwidth is the minimum bandwidth that has to be allocated to HRT and SRT services in order to guarantee each service operates normally. Without this minimum bandwidth, some of these services will terminate their end-to-end connections. Hence, we have

$$B_h=\sum_{i=1}^{|s_h|}R_h[\,i\,]+\sum_{j=1}^{|s_s|}R_{sm}[\,j\,] \tag{4}$$

where $|S_h|$ and $|S_s|$ are the number of services in sets S_h and S_s, respectively[2].

Soft allocated bandwidth (B_s) is the bandwidth that can be flexibly allocated to EBE and SRT services and is given by $B_s=\mu-B_h$.

Because B_h is already reserved, we only need to B_s to EBE and SRT services. For B_s, our allocation goals are 1) enabling as many SRT services as possible to operate at their target rates, 2) ensuring no SRT service operates at a higher rate than its target rate, and 3) allocating as much bandwidth as possible to EBE services and some SRT services with more soft required bandwidth. Let $U[\,i\,]=R_{st}[\,i\,]-R_{sm}[\,i\,]$ be the soft required bandwidth of SRT service i, and $\mathbf{U}=\{U[1], U[2], \ldots, U[|S_s|]\}$ be a non-descending vector $U[\,i\,]\leqslant U[\,i+1\,]$ ($i\in[1, |S_s|-1]$). To achieve our design goals, we firstly translate our bandwidth allocation to an optimization problem as follows:

$$\begin{aligned}&\max\ k\\&\textit{subject to}\quad 1\leqslant k\leqslant |S_s|\\&\qquad B_s=\sum_{i=1}^{k}U[\,i\,]+(|S_e|+|S_s|-k)\cdot x\\&\qquad U[k]\leqslant x,\end{aligned} \tag{5}$$

where k is the number of SRT services with less soft required bandwidth and operating at their target rates, and x is the amount of bandwidth allocated to an EBE service or a SRT service with more soft required bandwidth than any of the preceding k services. If k does not satisfy the constraints in (5), no SRT service can operate at its target rate, each of EBE and SRT services obtains the same soft allocated bandwidth $B_s/(|s_e|+|s_s|)$, and their rates are:

$$\begin{aligned}&r_e=B_s/(|s_e|+|s_s|),\\&r_s[i]=R_{sm}[i]+B_s/(|s_e|+|s_s|),\ \ i\in[1,|S_s|\,]\end{aligned} \tag{6}$$

Otherwise, if $k>0$ under the constraints in (5), the preceding k SRT services can operate at their target rates, and the rates of other services are

$$\begin{aligned}&r_e=\frac{B_s-\sum_{l=1}^{k}R_{st}[l\,]}{|S_e|+|S_s|-k},\\&r_s[i]=\begin{cases}R_{st}[i\,], & i\in[1,k]\\ R_{sm}[i]+\dfrac{B_s-\sum_{l=1}^{k}R_{st}[l]}{|S_e|+|S_s|-k}, & i\in[k,|S_s|\,]\end{cases}\end{aligned} \tag{7}$$

By allocating the latest throughput μ to traversing services, we avoid buffer overflow at the bottleneck RSU and use resources efficiently. In addition, the flexible bandwidth allocation scheme takes bandwidth requirements and fairness into account and improves QoS of different services. The pseudo codes of congestion avoidance procedure are described in Algorithm 1.

4.2 Avoiding Starvation

If the latest aggregate incoming rate λ is less than throughput μ, the buffer of the RSU starts to shrink, and

$$q-2T_{f\max}(\mu-\lambda)\leqslant Q_{\min} \tag{8}$$

where constant $Q_{\min}$ is the lower bound of its buffer size to avoid buffer starvation, the RSU should increase the aggregate incoming rate by allocating the latest throughput μ to different traversing flows for efficient resource utilization. As for the bandwidth allocation, the RSU uses the same allocation principle as (5), (6), and (7) to avoid starvation.

Assisted with the active bandwidth allocation mechanism, the RSU is able to reduce the probabilities of buffer overflow and starvation, which can not only reduce packet losses due to traffic congestion but also efficiently utilize the wireless resource. However, rate feedback packets may be lost due to channel error, noise, collisions, etc., and consequently wireline servers are unable to conduct rate adjustment accordingly due to the losses of feedback packets. In addition, feedback delay usually exists between the bottleneck RSU and wireline servers, and this means that sharp wireless bandwidth fluctuation during this interval leads to large oscillation in buffer size. As a result, buffer overflow and starvation cannot be entirely eliminated, and we should take a different approach to each of these

[2] For HRT services or some SRT services with minimum bandwidth requirement, the hard allocated bandwidth must be guaranteed by the network, otherwise the admission control procedure will be invoked to terminate some connections, which is out of the research scope of this paper. Therefore, we assume the throughput of the bottleneck RSU μ is always larger than the hard allocated bandwidth B_h.

cases.

Algorithm 1 Congestion Avoidance

compute parameters μ and λ;
$k = 0$;
if $\lambda > \mu$ **then**
  if $q + 2T_{f\max}(\lambda - \mu) \geqslant Q_{\max}$ **then**
    compute hard allocated bandwidth B_h;
    compute soft allocated bandwidth B_s;
    obtain parameter k;
    if $k = 0$ **then**
     calculate r_e, and feed it back to every EBE service;
     $i = 1$;
     while $i \leqslant |S_s|$ **do**
      calculate $r_s[i]$, and feed it back to SRT service i;
      i++;
     end while
    else
     calculate r_e, and feed it back to every EBE services;
     $i = 1$;
     while $i \leqslant k$ **do**
      feed target rate $R_{st}[i]$ back to SRT service i;
      i++;
     end while
     while $i \leqslant |S_s|$ **do**
      calculate $r_s[i]$, and feed it back to SRT service i;
      i++;
     end while
    End if
  End if

4.3 Buffer Overflow and Starvation

Buffer overflow and starvation cannot be completely eliminated due to feedback packet loss, feedback delay, and wireless bandwidth fluctuation. Therefore, we need to reduce congestion loss or starvation time in such situations. When the buffer overflows, the RSU should decrease the aggregate incoming rate to less than the link throughput. This makes the buffer shrink to a reasonable size as soon as possible. To achieve this design goal, the proposed VRC focuses on the soft allocated bandwidth B_s, and distributes the soft allocated bandwidth $\alpha \cdot B_s$ to EBE and SRT services, where $\alpha \in (0,1)$. As a result, the total allocated bandwidth $B_h + \alpha \cdot B_s$ is less than the current throughput $\mu = B_h + B_s$, which leads to a decrease in the buffer size. When the buffer size keeps decreasing and satisfies (8), the RSU allocates the actual bandwidth μ to traversing services. Similarly, when the buffer enters a starvation state, the RSU allocates $B_h + \beta \cdot B_s$ to traversing flows, where $\beta > 1$, and the buffer is not empty. If the buffer size continues to increase and satisfies (4), the actual bandwidth μ is allocated to traversing services. The RSU conforms to the same principles as (6) and (7) when conducting bandwidth allocation among traversing flows in cases of buffer overflow and starvation.

4.4 Implementation Issues

To efficiently implement the active bandwidth allocation mechanism, the proposed VRC scheme exploits coordination between end systems and the bottleneck RSU. When a mobile vehicle initiates a connection-oriented EBE service, the feedback rate r_e maintained at the associated RSU piggybacks in the transport layer ACK packet of TCP three-way handshake process (SYN/SYN-ACK/ACK), which is then delivered to the wireline server. Without tentatively probing the available bandwidth as the traditional slow-start process in TCP, the server starts data transmission at a rate r_e. On the other hand, an SRT service uses connectionless user datagram protocol (UDP) protocol at the transport layer, which means there are no control packets to take rate information. In VRC, the wireline server of a SRT service i starts data transmission with its target rate $R_{st}[i]$ when the end-to-end connection is initialized, and does not adjust its sending rate unless rate feedback information is returned. The RSU takes note of the initial packet arrival rate of service i as its target rate. In addition, for SRT service i of a specific type of multimedia services, its minimum guaranteed rate $R_{sm}[i]$ is usually fixed and stored at the RSU for subsequent bandwidth allocation. Similarly, an HRT service i also initiates end-to-end data transmission at a constant rate $R_h[i]$, and the RSU regards the packet arrival rate of service i as its constant sending rate.

Furthermore, when the bottleneck RSU needs to allocate bandwidth in order to avoid buffer starvation and congestion or to react to buffer starvation and overflow, it returns the allocated rates to wireline servers directly through rate feedback packets. In some references [17], the bottleneck router inserts feedback rates into the headers of data packets and delivers them to receivers. Then, receivers return ACK packets carrying the rate information to source nodes. The proposed VRC scheme takes a cross-layer approach, and the allocated rates are returned to servers directly from the bottleneck RSU within the wireline domain. This is due to the following considerations. On the one hand, the loss of rate adjustment information probably invalidates our bandwidth allocation mechanism, which may lead to buffer overflow or starvation at the bottleneck RSU. Direct packet exchange within the wireline domain avoids wireless transmission of rate feedback information from mobile users to the RSU and greatly increases the transmission reliability in a hostile V2I environment. On the other hand, large and unstable feedback delay usually makes the rate control scheme insensitive to the dynamic communication environment. This tends to degrade network performance (e.g., underutilization of the wireless channel) [18]. However, in the stable wireline domain, direct rate feedbacks from the intermediate RSU to the servers significantly reduce feedback delay and also make the delay predictable. This prevents performance degradation caused by network dynamics. We do not demonstrate the format of rate feedback packet in this paper.

5 Simulation Results

We implemented the proposed VRC scheme in network simulator (NS)-2 [19] and replaced the AIMD mechanism with VRC rate-adjustment mechanism to support EBE services in Freeze-TCP [20]. In addition, we integrated VRC rate-control mechanism into TCP friendly rate control (TFRC) [21] to support SRT services. In the simulated V2I communication network, vehicles traveled on a two-lane road with randomly selected speeds of between 5~25 m/s, and started data downloading from wireline servers through their associated RSUs. As an edge router, an RSU relays diverse types of multimedia traffic to mobile vehicles within its transmission range. This usually leads to traffic congestion at the bottleneck roadside unit in V2I communications. The parameters adopted in the simulations are shown in **Table 1.**

5.1 Average Download Packets

For EBE services in V2I communication networks, one of the most important performance metrics is the amount of data packets received during the sojourn time of a vehicle within the associated RSU. We first compare the average number of downloaded packets of the proposed vehicular rate control with that of AIMD mechanism by increasing the number of EBE services (**Fig. 2a**). The number of received data packets decreases as the number of active flows increases for both schemes. However, the average number of downloaded packets using the proposed scheme is much higher than that using AIMD. In VRC, when a vehicle sets up end-to-end connection after moving into an RSU, the vehicle immediately obtains a fair share of the allocated rate that is stored at the RSU. By avoiding congestion, the presented VRC not only avoids packet loss caused by buffer overflow, and by avoiding starvation, the buffer is not idle. This results in efficient use of the wireless channel resource. However, for EBE services with AIMD scheme, the wireline source nodes have to tentatively probe the available bandwidth by gradually increasing their sending windows, and are unable to fully utilize the wireless channel resource quickly.

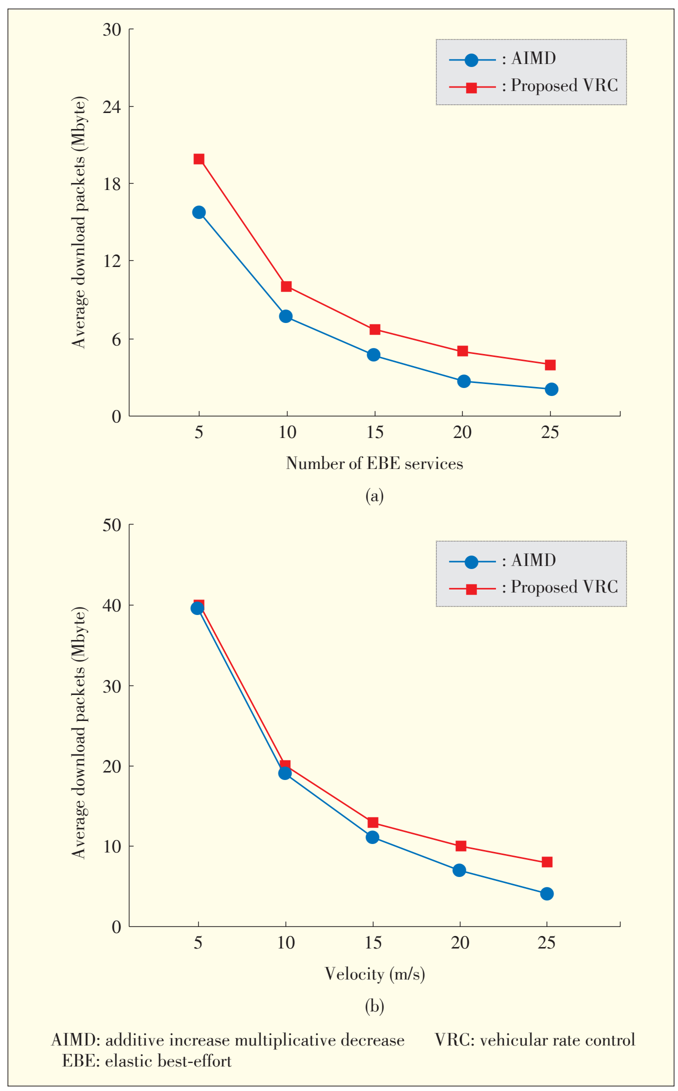
▼Table 1. Parameters in simulations

Parameter	Value	Parameter	Value
MAC	802.11DCF	Queue	Droptail
α	0.8	β	2
$T_{f\max}$	10 ms	T	100 ms
Basic rate	1 M	Data rate	11 M
R	250 m	DATA packet	512 byte
Buffer size	100 (packet)	$Q_{\max}$	99 (packet)
$Q_{\min}$	1 (packet)	$R_h[i]$	50 kb/s
$R_{st}[i]$	50 kb/s	$R_{sm}[i]$	5 kb/s

We initiate 10 EBE services, and compare the average download packets between VRC and AIMD by gradually increasing the moving velocity of vehicles (**Fig. 2b**). The average number of downloaded packets in both schemes reduces when vehicles travel at higher speeds. However, the presented VRC performs better than AIMD in each case because when the sojourn time reduces, the ratio of time spent in the slow start period increases for AIMD. This greatly decreases the number of download packets. However, the number of download packets with VRC decreases only because the reduced sojourn time caused by increasing travel speeds of vehicles.

5.2 Average Inter-Arrival Delay

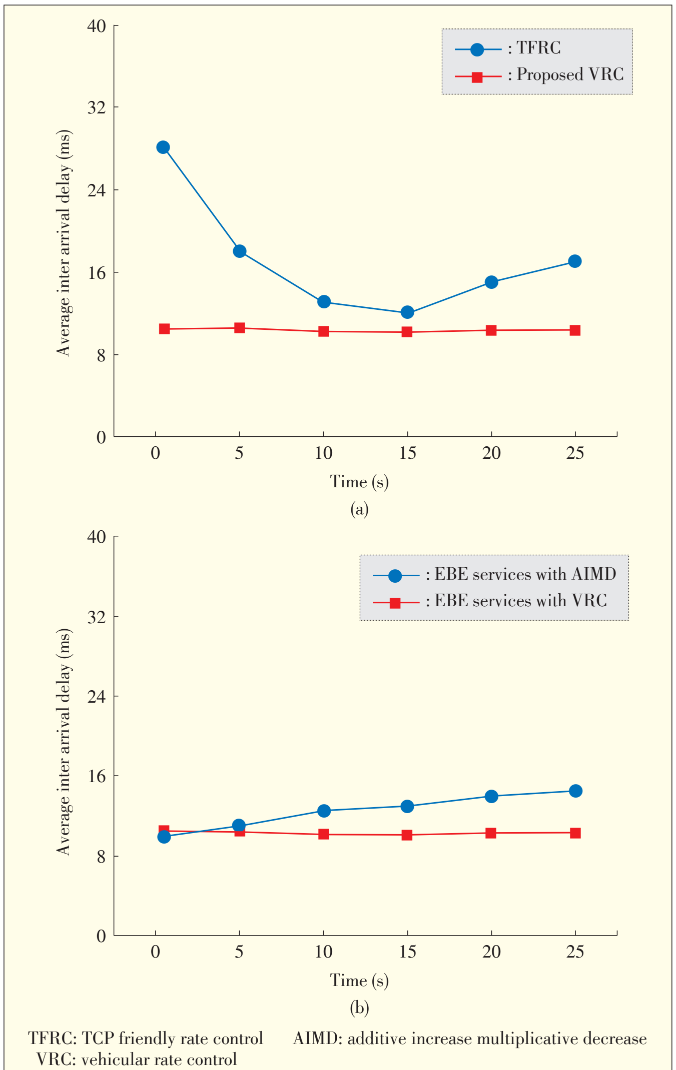
For multimedia real-time services, packet inter-arrival delay at mobile vehicles is a crucial performance metric. We set up five soft real-time services and 10 EBE services and then compare inter-arrival delay of SRT services between VRC and



▲Figure 2. Performance comparisons of average download packets.

TFRC during the sojourn time of vehicles. The delay of TFRC is much higher than that of the proposed VRC in the beginning (**Fig. 3a**); however, it decreases slowly and then increases in the end as sojourn time elapses. With the proposed VRC scheme, the inter-arrival delay of SRT services is relatively constant throughout the sojourn. In VRC, different types of traversing flows operate at higher sending rates to fully utilize the wireless channel resource after they travel into the coverage of an RSU. This enables a lower and steadier inter-arrival delay. TFRC has the slow-start mechanism and gradually increases its sending rate with an initial rate of one packet per round trip time (RTT). TFTC also decreases sending rate because of different kinds of packet loss during the vehicle sojourn time. This makes the inter-arrival delay fluctuate.

In **Fig. 3b**, we initiate 5 HRT services with constant sending rates and 10 competing EBE services. We compare inter-arrival delays of HRT services with EBE services controlled by VRC and AIMD, respectively. At the beginning, the delay for HRT services with AIMD is lower than that for our proposed scheme (Fig. 3a); however, this delay later exceeds it. This is because the queuing delay of HRT packets at the bottleneck RSU is short since AIMD based EBE flows initially operate at the slow start procedure, but greatly goes up when competing EBE flows operate at congestion avoidance process. However, EBE services operate at the fair allocated rates in the proposed VRC, and the queuing delay of HRT packets is relatively stable during the sojourn time of vehicles.



▲**Figure 3. Performance comparisons of average inter arrival delay.**

6 Conclusions

In this paper, we have proposed a cross-layer rate control scheme to support multimedia services in V2I networks. By tracking the latest throughput, aggregate arrival rate, and buffer size at the bottleneck RSU, source sending rates can be adjusted to avoid buffer overflow or starvation at the RSU. In addition, bandwidth can be efficiently allocated to different types of multimedia services by taking bandwidth requirements of each service into account. Simulation results show that the proposed VRC not only improves system throughput and efficiently utilizes the wireless channel resource but also meets different QoS requirements of multimedia services in V2I networks.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61101121; National High Technology Research and Development Program under Grant No. 2013AA102505; Key Laboratory Project Funds of Shenyang Ligong University under Grant No. 4771004kfs03; Zhejiang Provincial Natural Science Foundation of China under Grant No. LY12F01021; Educational Committee of Liaoning Province science and technology research projects under Grant No. L2013096.

References

[1] J. Eriksson, H. Balakrishnan, and S. Madden. "Cabernet: vehicular content delivery using WiFi," in *Proc. of ACM MOBICOM' 08*, San Francisco, 2008, pp. 199-210.

[2] Y. Bi, L. X. Cai, X. Shen, and H. Zhao. "Medium access control for QoS provisioning in vehicle-to-infrastructure communication networks," *ACM Mobile Netw. Appl.*, vol. 18, no. 2, pp. 174-185, Apr. 2013. doi: 10.1007/s11036-012-0350-x.

[3] L. Cai, X. Shen, J. W. Mark, and J. Pan, "QoS support in wireless/wired networks using the TCP-friendly AIMD protocol," *IEEE Trans. Wirel. Commun.*, vol. 6, no. 2, pp. 469-480, Feb. 2006.

[4] S. Floyd and E. Kohler. "Profile for datagram congestion control protocol (DCCP) congestion control ID 2: TCP-like congestion control," IETF, RFC 4341, Mar. 2006.

[5] A. Sridharan and B. Krishnamachari, "Explicit and precise rate control for wireless sensor networks," in *Proc. of ACM SenSys' 09*, Berkeley, CA, USA, 2009, pp. 29-42. doi: 10.1145/1644038.1644042.

[6] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. of ACM SIGCOMM' 02*, Pittsburgh, Pennsylvania, USA, 2002, pp. 89-102. doi: 10.1145/964725.633035.

[7] Y. Su and T. Gross, "WXCP: explicit congestion control for wireless multi-hop networks," in *Proc. of IFIP IWQoS' 05*, Passau, Germany, 2005, pp. 313-326. doi: 10.1007/11499169_25.
[8] F. Abrantes, J. Taveira Araujo, and M. Ricardo, "Explicit congestion control algorithms for time varying capacity media," *IEEE Trans. Mob. Comput.*, vol. 10, no. 1, pp. 81-93, Jan. 2011. doi: 10.1109/TMC.2010.143.
[9] N. Dukkipati, M. Kobayashi, R. Z. Shen, and N. McKeown, "Processor sharing flows in the internet," in *Proc. of IFIP IWQoS' 05*, Passau, Germany, 2005, pp. 267-281. doi: 10.1007/11499169_22.
[10] L. Barreto and S. Sargento, "TCP, XCP and RCP in wireless mesh networks: an evaluation study," in *Proc. of IEEE ISCC' 10*, Riccione, Italy, 2010, pp. 351-357.
[11] L. Barreto and S. Sargento, "XCP-Winf and RCP-Winf: congestion control techniques for wireless mesh networks," in *Proc. of IEEE ICC' 11*, Kyoto, Japan, 2011, pp. 1-6.
[12] W. H. Wang, M. Palaniswami, and S. H. Low, "Application-oriented flow control: fundamentals, algorithms and fairness," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1282-1291, Dec. 2006. doi: 10.1109/TNET.2006.886318.
[13] J. Jin, A. Sridharan, B. Krishnamachari, and M. Palaniswami, "Handling inelastic traffic in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1105-1115, Sept. 2010. doi: 10.1109/JSAC.2010.100915.
[14] J. Pu and M. Hamdi, "Utility-based fair bandwidth sharing in vehicular networks (extended)," *Wirel. Commun. Mob. Comput.*, vol. 10, no. 2, pp. 1648-1655, Dec. 2010. doi: 10.1002/wcm.1069.
[15] N. B. Yoma, J. Hood, and C. Busso, "A real-time protocol for the internet based on the least mean square algorithm," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 174-184, Feb. 2004. doi: 10.1109/TMM.2003.819582.
[16] Y. Bi, X. Shen, and H. Zhao, "Explicit rate based transmission control in vehicle to infrastructure communications," in *Proc. of IEEE ICCC' 13*, Xi'an, Shaanxi, China, 2013, pp. 704-709. doi: 10.1109/ICCChina.2013.6671202.
[17] K. Sundaresan, K. Sundaresan, H. Y. Hsieh, and R. Sivakumar, "ATP: a reliable transport protocol for ad hoc networks," *IEEE Trans. Mob. Comput.*, vol. 4, no. 6, pp. 588-603, Nov. 2005. doi: 10.1109/TMC.2005.81.
[18] W. Jiang, F. Ren, C. Lin, I. Stojmenovic, "Analysis of backward congestion notification with delay for enhanced ethernet networks," In *Proc. of INFOCOM' 12*, Orlando, Fl, USA, 2012, pp. 2961- 2965. doi: 10.1109/INFCOM.2012.6195738.
[19] "The Network Simulator-NS-2,"[Online]. Available: http://www.isi.edu/nsnam/ns/
[20] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. of INFOCOM' 00*, Tel Aviv, Israel, 2000, pp. 1537- 1545. doi: 10.1109/INFCOM.2000.832552.
[21] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. of ACM SIGCOMM' 00*, Stockholm, Sweden, 2000, pp. 43-56.

Manuscript received: February 17, 2014

Biographies

Yuanguo Bi (biyuanguo@ise.neu.edu.cn) is an associate professor at the College of Information Science and Engineering, Northeastern University, China. He received his BSc degree from Liaoning University, China, in 2003. He received his MSc and PhD degrees in computer science from Northeastern University, China, in 2006 and 2010. From 2007 to 2009, he was a visiting PhD student at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His current research interests include medium-access control, QoS routing, multihop broadcast, transmission control in vehicular networks.

Hangguan Shan (hshan@zju.edu.cn) received his BSc degree in electrical engineering from Zhejiang University in 2004. He received his PhD degree in electrical engineering from Fudan University in 2009. From 2009 to 2010, he was a postdoctoral research fellow at University of Waterloo, Canada. In February 2011, he joined the Department of Information Science and Electronic Engineering, Zhejiang University, as an assistant professor. He is the co-recipient of the Best Industry Paper Award from IEEE Wireless Communications and Networking Conference (WCNC) 2011. His current research interests include resource management and QoS/QoE provisioning in diverse wireless networks.

Xuemin (Sherman) Shen (xshen@bbcr.uwaterloo.ca) is a professor in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He received his BSc degree from Dalian Maritime University, China, in 1982. He received his MSc (1987) degree and PhD degree (1990) in electrical engineering from Rutgers University, USA, in 1987 and 1990. He was the associate chair for graduate studies at Rutgers from 2004 to 2008. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, wireless body area networks, vehicular ad hoc and sensor networks. He has co-authored and/or edited six books and has published more than 600 papers and book chapters on wireless communications and networks, control and filtering.

Hai Zhao (zhaohai@ise.neu.edu.cn) is a professor in the Department of Computer Science and Technology, Northeastern University, China. He received his BSc degree in electrical engineering from Dalian Maritime University, China, in 1982. He received his MSc and PhD degrees in computer science from Northeastern University, China, in 1987 and 1995. He is the director of Liaoning Provincial Key Laboratory of Embedded Technology. His current research interests include embedded Internet technology, wireless sensor network, pervasive computing, operating system, data and information fusion, and computer simulation and virtual reality. He has held programs such as National Natural Science Foundation of China, National High Technology Research and Development Program of China, and National Class Lighted Torch Plan. He has published more than 300 academic papers, four books, and one national standard. He has successfully applied for 10 patents and received six awards for science and technology from the Liaoning province and ministry of China. He received allowance of the State Council due to his special contributions to the development of education.

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 002
http://www.cnki.net/kcms/detail/34.1294.TN.20140312.1134.001.html, published online March 12, 2014

Advanced Leader Election for Virtual Traffic Lights

Florian Hagenauer, Patrick Baldemaier, Falko Dressler, and Christoph Sommer
(Institute of Computer Science, University of Innsbruck, Technikerstr. 21a, 6020 Innsbruck, Austria)

Abstract

We examine the network performance of algorithms for self-organized traffic management. In particular, we focus on wireless networking between cars. One of many technologies that make road traffic safer and more efficient is the Virtual Traffic Light (VTL) system, which is able to coordinate the traffic flow at intersections without the need for physical lights. VTL takes a leading vehicle at an intersection and uses it to control the traffic lights. We developed algorithms for leader election and traffic light computation in realistic vehicular networking scenarios. Our key contribution is the extension of this algorithm to support arbitrary intersection layouts. We investigated the proposal in synthetic and realistic scenarios. The results show that, overall, VTLs use network resources efficiently and positively influences driving experience. It performs better than stationary traffic lights for a low to medium network load. We also identify potential optimizations to deal with high network load and to improve fairness.

Keywords

virtual traffic lights; self-organization; vehicular networks

1 Introduction

A way to safer, more efficient traffic is hinted at in the 2010 report *Towards a computational transportation science* [1]. This report introduces a new scientific discipline in which the work of computer scientists and transportation experts is combined to produce more useful models and technologies for future transportation.

Safety at intersections has become a key research challenge within the inter-vehicle communication (IVC) community. Solutions range from intersection warnings to more intelligent assistant systems [2]–[5].

In the U.S., more than 240,000 traffic lights regulate intersections [6]. Of these, more than 50% are in need of repair [7]. Others are running at suboptimal levels, leading to raffic delays of up to 40% in excess of what they would normally be if the lights were optimally configured [6]. Repairing and/or reconfiguring these traffic lights is costly, and many intersections (up to 99.5% in the U.S. [8]) are simply left without traffic lights.

A completely new approach to complementing (or even replacing) physical traffic lights is the Virtual Traffic Light (VTL) system [8]–[10]. Vehicles approaching an intersection exchange messages wirelessly and cooperate to create a dynamic traffic light program for the junction. This information is then presented to the driver on an in-vehicle display; thus, physical traffic lights are replaced.

The VTL approach offers numerous benefits, including eliminating the cost of deploying traffic light infrastructure on every street. VTLs can also react quicker than normal traffic lights to microscopic traffic conditions.

One of many different wireless technologies that VTL can be built on is the IEEE 802.11p standard [11]. Based on this, IEEE DSRC/WAVE and ETSI ITS G5 are being standardized for vehicular networking. We are now in an era of radical change in terms of car manufacturing and road traffic management. Such changes are supported by the U.S. federal government's National Highway Traffic Safety Administration (NHTSA), which announced in February 2014 that it plans to require all new cars to have wireless technology that broadcasts the car's location, speed, direction, and other data so that drivers can be warned of an impending collision [12]. This announcement coincides with the final standardization of higher-layer networking protocols in Europe by the European Telecommunications Standards Institute (ETSI).

In this paper, we extend the model we developed in [13]. This allows us to investigate the feasibility of using current vehicular-networking proposals to realize VTL on a larger scale. In particular, we focus on the networking aspect and investigate the feasibility of establishing both the VTL and car clusters in a reliable way. Our new leader-election and traffic-light-computation algorithm now supports arbitrary intersection layouts. We investigated this extended algorithm in a realistic scenario. We found that VTL uses network resources efficiently and positively impacts the driving experience. VTL performs better than stationary traffic lights for a low to medium network load. We also identified optimization potential to deal with high network load and improve fairness.

2 Related Work

VTLs have been repeatedly worked towards in the literature. One of the first examples is by Dresner et al. [10], which describes a system that allows vehicles to wirelessly coordinate when they pass at intersections. This eliminates the need for physical traffic lights. Using a custom-built road traffic simulator, the authors show a large increase in efficiency.

Gradinescu et al. [9] describe adaptive traffic lights that gather data from approaching cars and adapt green time. However, this research is primarily concerned with physical traffic lights. Using a custom built discrete-event simulation tool, they show that the average delay of vehicles decreases significantly if traffic lights can make use of wirelessly exchanged information.

Avin et al. [14] introduce a scheme to place another traffic light in front of intersections. These traffic lights should be dynamically placed and synchronized to allow a vehicle to pass through an intersection much quicker. To ease repositioning traffic lights virtual counterparts are introduced. From microscopic road traffic simulation using SUMO, Avin et al. show that this scheme reduces the number of delayed vehicles by up to 20%.

The most complete proposal for VTLs to date, and the one that we build on, is that of Ferreira et al. [8]. By running simulations, they show that VTLs have the potential to reduce CO_2 emissions by up to 18% [15]. Simulations were run using DIVERT in conjunction with NS-3. The impact of different human-machine interfaces for VTLs was investigated by Olaverri-Monreal et al. [16]. The authors showed that brake activity when VTLs are used is similar to brake activity when physical traffic lights are used. Viriyasitavat and Tonguz [17] ran simulations on SUMO to show that emergency vehicles can reach their destination minutes earlier when VTLs are used.

We complement these studies by using a state-of-the-art combined network and traffic-simulation environment. The Veins framework connects the well-established network simulator OMNeT++ with the detailed vehicular simulation SUMO [18]. In our earlier work, we already developed algorithms for leader election and traffic light computation in realistic vehicular networking scenarios [13]. Our main contribution in this paper is the extension of this algorithm to support arbitrary intersection layouts and to investigate its performance in more realistic simulations.

3 Architecture

3.1 General Architecture

The basic idea for the VTL algorithm is proposed by Ferreira et al. [8]. To support a system using the algorithms, vehicles have to have a wireless networking device and the means of determining their current position. IEEE 802.11p and GPS support these needs and can be used to make VTL work. GPS is already used in many vehicles nowadays, and it can be safely assumed that cars are equipped with it. Dedicated short-range communications (DSRC) technologies are not widely deployed yet, but the recent NHSTA announcment paves the way for mandating this technology in the U.S. [12].

The following occurs when VTL is active at an intersection:

1) Vehicles broadcast data about their current position to other cars in close proximity. In the future, other information, such as speed and acceleration, may be broadcast. The data is stored in a neighbor table and is updated if newer information is received. It is envisioned that such information will be automatically provided by basic services of vehicular networks, e.g., local dynamic maps (LDMs) maintained by cooperative awareness messages (CAMs) in ETSI ITS G5.
2) If a crossing conflict is detected at an intersection, a virtual traffic light has to be calculated. Such a conflict occurs when two vehicles approach an intersection and a collision is imminent. In this case, a leader needs to be selected to determine the traffic light program.
3) The vehicles select a single car to be the leader at the intersection. This vehicle generates a traffic light program and sends the information to the other vehicles.
4) Vehicles receive traffic light data that is displayed inside the vehicle to the driver.
5) If the traffic light program changes, the leader sends an update.
6) When the leader crosses the intersection, a new leader has to be selected. This can be done by the old leader automatically selecting a leader or by initiating a new election. In this case, the algorithm continues at step 3.

An early work describing VTL approaches [8] provides a very good introduction to the general VTL concept, but detailed explanation about leader election is lacking. In [8], a simple table-based scheme is used for traffic-light program calculation. In [13] we addressed these issues and proposed a detailed leader-election scheme and a more dynamic traffic light program in which a single vehicle announces an election and others reply with their distance to the intersection. After a timeout, the vehicle closest to the intersection is selected as the new leader and determines the new traffic light program. If multiple elections are initiated by different vehicles, every vehicle has a single, comparable ID to break ties. This may be, for example, the network address. For a more detailed description of the algorithm, we refer to [13]. The used model had two deficiencies, which we address in this paper:

1) A simulated physical traffic light was used by the road traffic simulation and was controlled by the VTL leader. This meant that the traffic light program was directly obeyed in the road traffic simulation without the need to send it to

each vehicle approaching the intersection. Therefore, it was not possible to simulate effects introduced by multiple leaders.

2) The algorithm was not aware of non-conflicting lanes. Such lanes can be given a green light without the possibility of any crashes. With our first model, only a single lane was given a green light while others had to wait. Such a scenario produces unnecessary delay for the drivers.

3.2 Improved Features

After analyzing the problems with our first proposal, we started to develop an updated version that solves these problems. The first thing we did was to remove the dependency from SUMO traffic lights. We configured SUMO so that it always keeps all traffic lights red and forces vehicles to stop. We selectively allowed any vehicles with a VTL state of green to ignore the traffic light. With this in place, cars only followed a local traffic light state table that stores information about the next intersection. Until a traffic light broadcast is received, a vehicle assumes the state unknown for a VTL. If this happens, the vehicles stop at the intersection. In this case, a timeout is initiated, and afterwards, the current state is requested by the intersection's leader. This is done to prevent vehicles waiting unnecessarily at the intersection.

Second, it is now possible to give a green light to multiple non-conflicting lanes. Our proposed protocol for VTL can detect such lanes and give a green light to multiple lanes at the same time. This is done by selecting one lane as before and iterating over all other lanes, adding them to the set of lanes to be given a green to if there is no conflict.

Third, an update enables vehicles to store a simplified topology of the road network. In this view on the street network, lanes (which share the same endpoint and do not branch) are grouped together. This additional topology helps them quickly identify which (if any) is the next traffic light that VTL needs to be performed for.

Finally, we further improved the election algorithm presented in [13], by adding fast leader hand-off: when a VTL leader crosses an intersection, it preferably hands-off the task of leader to a vehicle that is stopped in order to reduce rapid leader switching.

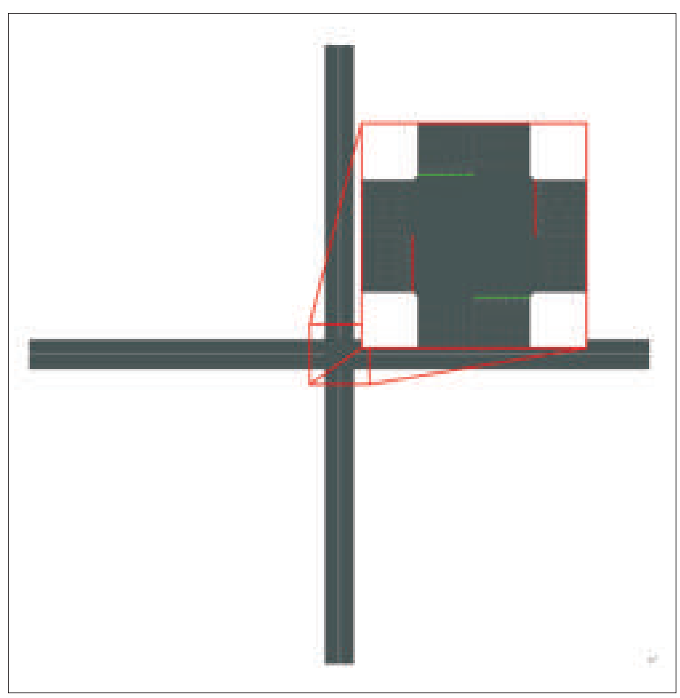
4 Evaluation

We evaluated the performance of our proposed VTL protocol in a synthetic scenario and a realistic scenario.

Our synthetic scenario involved a simple four-way intersection in which each road was 400 m long. To force the intersection to have a very high load without substantially hindering traffic flow, we made each road twelve lanes wide (six lanes each direction) (**Fig. 1**) and forced vehicles to only go straight through the intersection at a speed limit of 50 km/h. For the same reason, we did not include buildings, which would obstruct communication between vehicles. This scenario enabled us to study VTL with arbitrary, uniform traffic flows. We picked a range of 225 to 2700 vehicles per hour passing through the intersection and investigated the impact of network load on VTL performance.

The first metric we used in this investigation was mean total travel time, that is, how long a vehicle took on average to pass through the intersection. This metric included the time it took for a vehicle to travel the 400 m to the intersection and the time it took to travel the 400 m after it. We compared travel times for an intersection managed by conventional traffic lights and for the same intersection managed by our proposed protocol extending VTL. With the conventional traffic lights, we configured standard fixed durations of 31 s green, 6 s yellow, and 37 s red.

Fig. 2 shows the results for seven different traffic densities. It shows the mean value across all repetitions and the 95% confidence interval of the mean. When traffic density is low, average travel time is improved if the intersection is managed with VTL. When conventional traffic lights are used to regulate traffic, cars take up to 20% longer to reach their destination. However, when traffic density is higher, this benefit becomes less pronounced. In fact, VTL and conventional traffic lights perform at about the same level when 1800 vehicles per hour pass through the intersection. VTL performs slightly worse than conventional traffic lights for higher-density traffic. This is not because the intersection cannot handle this amount of traffic. For the simulated densities, the mean travel time does not vary with respect to traffic density if the intersection is managed by conventional traffic lights. Therefore, the reduced effectiveness



▲Figure 1. Synthetic simulation scenario: multi-lane four-way intersection.

of VTL must be due to some other effect.

We therefore investigate a second metric which is indicative of the system network performance: the channel access success ratio. This metric shows how often the channel is found to be idle when trying to send a packet (as opposed to finding the channel busy and triggering a back-off).

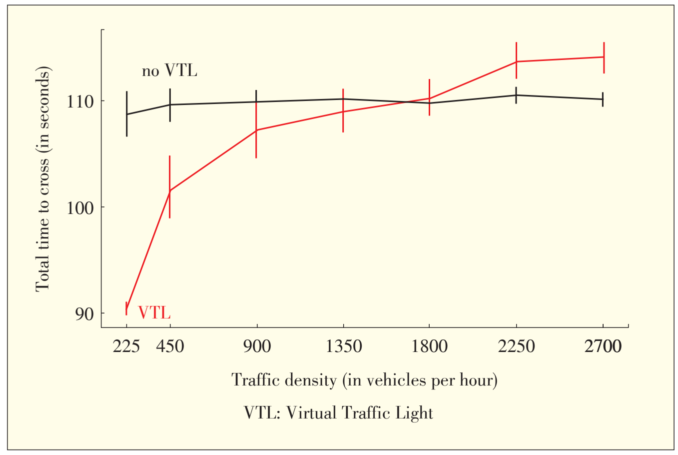
Fig. 3 shows the mean value of this metric as traffic density increases. The probability of finding the channel empty decreases linearly as traffic density increases. This probability decreases from almost 100% for 225 vehicles per hour to just above 90% for 1800 vehicles per hour and approximately 85% for 2700 vehicles per hour. Taken together with previous results, this means that leader election and traffic-light-program dissemination are very efficient (after all, the decrease is only linear). Still, even with as few as 10% failed channel access attempts, the proposed protocol is already no more efficient than conventional traffic lights.

This points to various avenues for future work. While it can be argued that VTLs are unlikely to be needed in regions with high traffic densities, the results point to a strong relationship between available network capacity and VTL performance.

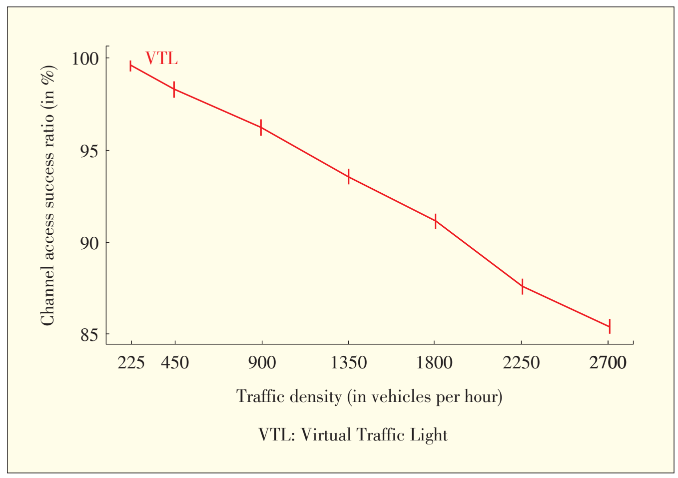
For now, we restrict our analysis to favorable network conditions; that is, low traffic density where the full network capacity is available to the VTL system. We turn to the potential impact of VTL on traffic within a city. Rather than considering a scenario with one simple intersection, we now simulate VTL in a more realistic traffic scenario.

The realistic scenario is based on real geodata from OpenStreetMap of Ingolstadt, Germany. The data accurately reflects road and building topology and geometry, speed limits, right of way, and one-way streets. We took great care to accurately model intersection topologies; that is, we added correct turning lanes, configured traffic signal timing, and added buildings that attenuate radio propagation. As in earlier work [19], we simulated traffic throughout the whole of Ingolstadt, but only investigated nodes driving within a 1.5 km^2 Region of Interest (ROI) (**Fig. 4**). This ROI contains a typical mix of high-capacity and low-capacity roads, traffic lights, and unregulated intersections as well as high and low node density areas. We generated traffic flows according to low demand, an average of 25 vehicles per square kilometer. This realistic scenario enabled us to get an idea of the impact of the designed VTL system on urban traffic flow. We compared a city managed by conventional traffic lights to the same city managed entirely with VTLs. In the interest of fairness, we only used VTLs on those seven intersections that used to have conventional traffic lights. All other intersections remained unmanaged with vehicles following the city's right of way rules (yield to priority road or yield to right).

Fig. 5 shows the results for the first metric we in-

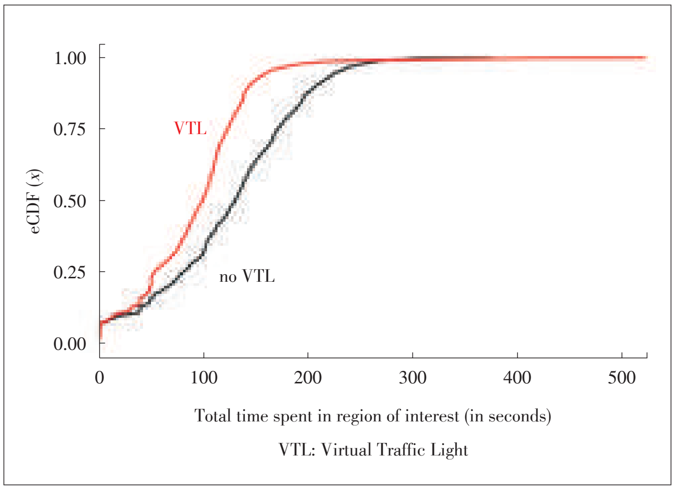


▲**Figure 2. Synthetic scenario: traffic density vs. travel time.**



▲**Figure 3. Synthetic scenario: traffic density vs. ratio of successful channel accesses (out of all packets sent).**

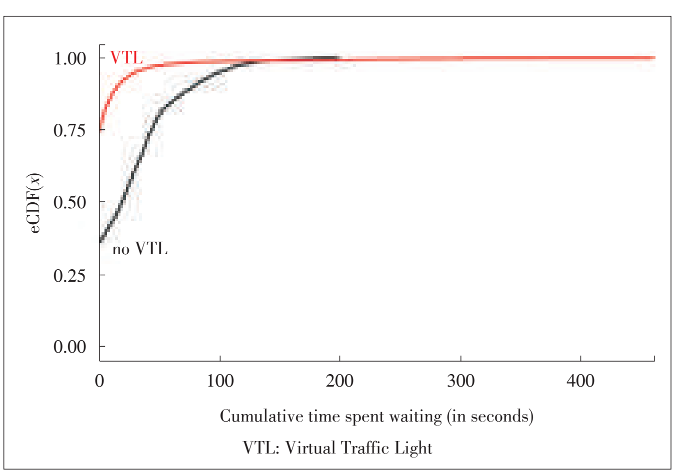
▲**Figure 4. Realistic simulation scenario: 1.5 km^2 Region of Interest (ROI) in the city of Ingolstadt.**



▲**Figure 5. Ingolstadt: travel time.**

vestigated; that is, the total time vehicles took to reach their destination (or to leave the ROI). The results are plotted as an empirical cumulative density function (eCDF) over all simulation runs to show the actual distribution. Both distributions have the same features: travel times are either extremely short (resembling vehicles that only briefly passed through a small part of the ROI) or are approximately normal, distributed around approximately 110 s and 130 s (mirroring the 20% difference observed in our simulations of a single intersection). There is also a pronounced right tail; that is, some vehicles are substantially slower, particularly in the scenario where vehicles relied on VTL to coordinate crossing.

To determine the reason for this, we investigate a new metric: the cumulative time that each vehicle spent waiting, that is, moving less than 0.1 m/s (0.36 km/h) (**Fig. 6**). The proportion of vehicles that did not have to wait at all increased from 36% to 74% when some intersections were managed with VTLs, and overall waiting times decreased substantially. Yet this metric also explains the reason for the pronounced right tail in the distribution of travel times. A very small number



▲**Figure 6. Ingolstadt: waiting time.**

(0.7%) of vehicles had to wait twice as long as even the longest waiting time incurred when conventional traffic lights were used. Closer investigation reveals that these vehicles are (unfortunately) VTL leaders on unfrequented streets. It took a long time for enough vehicles to queue behind the leader on the unfrequented street to outweigh the amount of traffic on the major road. Therefore, our naïve VTL traffic light program caused major delays for these vehicles. Although traffic flow was substantially smoother overall, this 0.7% of drivers might feel unfairly treated by VTL.

5 Conclusion

In this paper, we evaluated an improved VTL system that can manage arbitrary intersection geometries. Using two scenarios—a realistic urban road network and a synthetic single intersection—we investigated the impact of VTLs on traffic flow within a city as well as the overall network performance of VTLs.

We showed that, overall, VTLs positively affected the driving experience of our simulated vehicles, cutting total travel time (by approximately 20%) and wait times (74% of vehicles did not have to wait at all). However, we also identified optimization potential: a few vehicles had significantly longer wait times. This points to a lack of consideration of fairness in the proposed algorithm. We further showed that our proposed protocol for VTL uses network resources efficiently: channel load scales linearly with traffic density. Still, a deterioration of its performance in scenarios with than 10% of channel load points to optimization potential in this area as well.

Future work might also include improved optimizations of which lanes to give green to, exploiting the full potential of combining multiple non-conflicting lanes into one green phase.

References

[1] S. Winter, M. Sester, O. Wolfson, and G. Geers, "Towards a computational transportation science," *Journal of Spatial Information Science*, vol. 2, no. 1, pp. 119–126, 2011. doi: 10.1145/1942776.1942785.

[2] S.-H. Chang, C.-Y. Lin, C.-C. Hsu, C.-P. Fung, and J.-R. Hwang, "The effect of a collision warning system on the driving performance of young drivers at intersections," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 5, pp. 371 – 380, 2009. doi: 10.1016/j.trf.2009.05.001.

[3] H. Berndt, S. Wender, and K. Dietmayer, "Driver Braking Behavior during Intersection Approaches and Implications for Warning Strategies for Driver Assistant Systems," in *IEEE Intelligent Vehicles Symposium (IV'07)*, Istanbul, Turkey: IEEE, Jun. 2007, pp. 245–251. doi: 10.1109/IVS.2007.4290122.

[4] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, "To crash or not to crash: estimating its likelihood and potentials of beacon-based IVC systems," in *4th IEEE Vehicular Networking Conference (VNC 2012)*. Seoul, Korea: IEEE, Nov. 2012, pp. 25–32. doi: 10.1109/VNC.2012.6407441.

[5] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, "A ve-

hicular networking perspective on estimating vehicle collision probability at Intersections," *IEEE Transactions on Vehicular Technology*, 2014, to appear.
[6] "National traffic signal report card," National Transportation Operations Coalition (NTOC), Report 2012, 2012.
[7] P. Koonce, W. Kittelson, L. Rodegerdts, K. Lee, C. Swartz, J. Bonneson, P. Tarnoff, B. DeRoche, D. Bullock, and W. Tighe, "Traffic signal timing manual," U. S. Department of Transportation, Publication FHWA-HOP-08-024, Jun. 2008.
[8] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," in *7th ACM International Workshop on Vehicular Internetworking (VANET 2010)*, Chicago, IL: ACM, Sep. 2010, pp. 85–90. doi: 10.1145/1860058.1860077.
[9] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *65th IEEE Vehicular Technology Conference (VTC2007-Spring)*, Dublin, Ireland: IEEE, Apr. 2007, pp. 21–25. doi: 10.1109/VETECS.2007.17.
[10] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*. New York, NY: IEEE, Jul. 2004, pp. 530–537. doi: 10.1109/AAMAS.2004.190
[11] *Wireless Access for Vehicular Environments*, IEEE Draft Standard P802.11p/D4.0, Mar. 2008.
[12] M. L. Wald, "U.S. plans car-to-car warning system," *The New York Times*, p. B3, Feb. 4 2014.
[13] C. Sommer, F. Hagenauer, and F. Dressler, "A networking perspective on self-organizing intersection management," in *IEEE World Forum on Internet of Things (WF-IoT 2014)*. Seoul: IEEE, Mar. 2014.
[14] C. Avin, M. Borokhovich, Y. Haddad, and Z. Lotker, "Optimal virtual traffic light placement," in *8th International Workshop on Foundations of Mobile Computing (FOMC 2012)*. Madeira, Portugal: ACM, Jul. 2012. doi: 10.1145/2335470.2335476.
[15] M. Ferreira and P. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, no. 10, pp. 1–12, Oct. 2011. doi: 10.1109/TITS.2011.2169791.
[16] C. Olaverri-Monreal, P. Gomes, M. Silveria, and M. Ferreira, "In-vehicle virtual traffic lights: a graphical user interface," in *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*. Madrid, Spain: IEEE, Jun. 2012, pp. 471–476.
[17] W. Viriyasitavat and O. Tonguz, "Priority management of emergency vehicles at intersections using self-organized traffic control," in *Vehicular Technology Conference (VTC2012-Fall)*, Quebec City, Canada: IEEE, Sep. 2012, pp. 1–4. doi: 10.1109/VTCFall.2012.6399201.
[18] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3– 15, Jan. 2011. doi: 10.1109/TMC.2010.133.
[19] C. Sommer, D. Eckhoff, and F. Dressler, "IVC in cities: signal attenuation by buildings and how parked cars can improve the situation," *IEEE Transactions on Mobile Computing*, 2014, to appear.

Manuscript received: February 17, 2014

Biographies

Florian Hagenauer (hagenauer@ccs-labs.org) received his BSc and MSc degrees in computer science from the University of Innsbruck, Austria, in 2011 and 2013, respectively. In October 2013, he joined the Computer and Communication Systems Group to work on problems in Car-to-X communication.

Patrick Baldemaier (patrick.baldemaier@ccs-labs.org) received his BSc in computer science from the University of Innsbruck, Austria, in 2012. Since October 2013, he has been working on his master thesis "Simulation of Self-Organizing Intersection Management" at the Computer and Communication Systems Group.

Falko Dressler (dressler@ccs-labs.org) is a full professor of computer science and head of the Computer and Communication Systems Group at the Institute of Computer Science, University of Innsbruck. Dr. Dressler received his MSc and PhD degrees in computer science from the Department of Computer Science, University of Erlangen, in 1998 and 2003. He is an editor for *IEEE Trans. on Mobile Computing*, Elsevier *Ad Hoc Networks*, ACM/Springer *Wireless Networks (WINET)*, and Elsevier *Nano Communication Networks*. He regularly serves in the TPC of networking conferences such as IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. Dr. Dressler has written textbooks, including "Self-Organization in Sensor and Actor Networks," which was published by Wiley in 2007. Dr. Dressler is an IEEE Distinguished Lecturer in the fields of inter-vehicular communication, self-organization, and bio-inspired and nano-networking. Dr. Dressler is a Senior Member of the IEEE (COMSOC, CS, VTS) as well as a senior member of ACM (SIGMOBILE), and member of GI (KuVS). His research activities are focused on adaptive wireless networking and self-organization methods with applications in wireless ad hoc and sensor networks, inter-vehicular communication, bio-inspired and nano-networking, and network security.

Christoph Sommer (sommer@ccs-labs.org) received his PhD degree in engineering (Dr.-Ing., with distinction) and his MSc degree in computer science (Dipl.-Inf. Univ.) from the University of Erlangen in 2011 and 2006, respectively. In 2010, he was a visiting scholar with the research group of Ozan K. Tonguz at the Electrical and Computer Engineering Department of Carnegie Mellon University (CMU). In 2012, he was a visiting scholar with the research group of Mario Gerla at the Computer Science Department of the University of California, Los Angeles (UCLA). He is currently a postdoctoral researcher with the Computer and Communication Systems group at the University of Innsbruck. Since 2011, he has been a member of the ACM/Springer Wireless Networks (WINET) editorial board. His research is focused on questions regarding traffic efficiency and safety, as well as on security aspects of Car-to-X communication in heterogeneous environments.

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 003
http://www.cnki.net/kcms/detail/34.1294.TN.20140321.1538.001.html, published online March 21, 2014

Trajectory-Based Data Forwarding Schemes for Vehicular Networks

Jaehoon (Paul) Jeong[1], Tian He[2], and David H. C. Du[2]
(1. Department of Software, Sungkyunkwan University, Suwon 440-746, Republic of Korea;
2. Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA)

Abstract

This paper explains trajectory-based data forwarding schemes for multihop data delivery in vehicular networks where the trajectory is the GPS navigation path for driving in a road network. Nowadays, GPS-based navigation is popular with drivers either for efficient driving in unfamiliar road networks or for a better route, even in familiar road networks with heavy traffic. In this paper, we describe how to take advantage of vehicle trajectories in order to design data-forwarding schemes for information exchange in vehicular networks. The design of data-forwarding schemes takes into account not only the macro-scoped mobility of vehicular traffic statistics in road networks, but also the micro-scoped mobility of individual vehicle trajectories. This paper addresses the importance of vehicle trajectory in the design of multihop vehicle-to-infrastructure, infrastructure-to-vehicle, and vehicle-to-vehicle data forwarding schemes. First, we explain the modeling of packet delivery delay and vehicle travel delay in both a road segment and an end-to-end path in a road network. Second, we describe a state-of-the-art data forwarding scheme using vehicular traffic statistics for the estimation of the end-to-end delivery delay as a forwarding metric. Last, we describe two data forwarding schemes based on both vehicle trajectory and vehicular traffic statistics in a privacy-preserving manner.

Keywords

VANET; DSRC; vehicular networks; data forwarding; vehicle trajectory

1 Introduction

Vehicular ad hoc networks (VANETs) have been studied intensively in wireless communications between vehicles for the driving safety and efficiency in road networks [1]–[7]. Every year, many South Koreans die in road accidents, and the fatality rate is increasing [8]. VANET can reduce the fatality rate by allowing vehicles to communicate directly with each other and avoid collisions in road networks. Also, in the era of high oil prices, VANET can determine the most efficient route for a car to take according to the final destination and real-time traffic conditions [9]. A variety of automotive cloud services [10] can be envisioned for vehicles and drivers. Such services include intelligent navigation, safe driving, automatic update of automotive software, onboard diagnostics (OBD) [11], reporting for online diagnosis, and smartphone vehicular remote control.

VANET for driving safety and efficiency has been researched in earnest since dedicated short range communications (DSRC) was standardized as IEEE 802.11p in 2010 [12]–[14]. IEEE 802.11p is an extension of IEEE 802.11a and defines the vehicular network characteristics, such as high-speed mobility and high vehicle density in roadways. Another important trend in vehicular networks is GPS-based navigation (e.g., dedicated GPS navigation [15] and smartphone navigation [16]), which is commonly used by drivers to navigate in unfamiliar areas. It was expected that 300 million mobile devices would be equipped with GPS receivers in 2009 [17]. These cutting-edge technologies for DSRC and GPS navigation open the way for research into the utilization of vehicle trajectories to make data forwarding more efficient in vehicular networks.

Let us assume the following setting in a vehicular network: The Traffic Control Center (TCC) [18] is a central node that collects traffic statistics (e.g., vehicle inter-arrival rate and average speed per road segment) in a road network. The TCC also maintains the trajectory, current position, speed, and direction of an individual vehicle to track vehicles registered in the TCC. Access points (APs) are sparsely deployed as roadside units (RSUs) [19] and are interconnected in order to provide vehicles with connectivity to wired networks (e.g., the Internet) that lead to the TCC. APs have limited coverage because of the

This work was supported by Faculty Research Fund, Sungkyunkwan University, 2013 and by DGIST CPS Global Center. This work was also partly supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2012033347) and by the IT R&D program of MKE/KEIT (10041244, SmartTV 2.0 Software Platform).

sparse deployment of APs by the deployment cost, so the vehicular networks are Disruption Tolerant Networks (DTNs) such that vehicles adopt the forward-and-carry approach for the multihop data delivery in road networks. Using this forward-and-carry approach, many data forwarding schemes (such as VADD [4], Delay-bounded Routing [5] and SADV [6]) for the vehicular networks have been proposed so far. However, these schemes use only vehicular traffic statistics (e.g., vehicle arrival rate per road segment) to compute a forwarding metric, such as Expected Delivery Delay (EDD) from a packet source to a packet destination. Thus, this EDD is used to select a next-hop vehicle toward the packet destination.

Given vehicle trajectories as future navigation paths available through GPS-based navigation systems, three data forwarding schemes, i.e., Trajectory-Based Forwarding (TBD) [1], Trajectory-Based Statistical Forwarding (TSF) [2], and Trajectory-Based Multi-Anycast Forwarding (TMA) [3] have been proposed to take advantage of these vehicle trajectories for 1) the better computation of a forwarding metric called EDD and 2) the determination of a target point that is the rendezvous position of the packet and the destination vehicle. TMA [3] is an extension of TSF [2] for the multicast data delivery from AP to multicast group vehicles as packet destinations. In this paper, we focus on the unicast data delivery scheme with TBD [1] and TSF [2] rather than the multicast data delivery scheme with TMA [3]. Note that this paper is the refined version of our early magazine article in [20], explaining TBD [1] and TSF [2] from the forwarding design aspect.

The remainder of this paper is structured as follows. Section 2 is a literature review of vehicular networking. Section 3 describes the modeling of link delay, packet delivery delay, and vehicle travel delay. Section 4 describes a data-forwarding scheme called vehicle-assisted data delivery (VADD), which is based on vehicular traffic statistics [4], as well as two data-forwarding schemes, TBD [1] and TSF [2], which are based on vehicle trajectories. Section 5 analyzes these two trajectory-based forwarding schemes along with VADD. Section 6 concludes the paper and describes future work.

2 Related Work

Much research has been done on multihop Vehicle-to-Infrastructure (V2I) [1], [4], [5], Infrastructure-to-Vehicle (I2V) [2], and Vehicle-to-Vehicle (V2V) [2] data-forwarding for safety and efficiency in vehicular networks. For these networks, VANETs are used for data forwarding over multihop toward the packet destination. VANET is different from traditional mobile ad hoc networks (MANETs) [20] because it supports the networking in road networks with layout rather than in two-dimensional open space assumed by MANETs. VANET is designed to take into account 1) high-speed vehicular mobility on roadways, 2) confined vehicular mobility on roadways, and 3) predictable vehicle mobility through roadmaps. Because of the first characteristic, there is frequent network partitioning and merging, so the forward-and-carry approach [1] is required instead of connection-oriented route usually used in MANET [21]. Because of the second characteristic, vehicular traffic statistics, such as vehicle arrival rate and average speed per road segment and vehicle branch probability at each intersection, can be collected [1]. The third characteristic is due to vehicle trajectory provided by GPS navigator [2].

Many data-forwarding schemes have been proposed with digital roadmaps and vehicular traffic statistics [4]–[6]. VADD [4] formulates the data-forwarding process as a stochastic process in road segments or at intersections, and is designed to minimize delivery delay. Delay-bounded routing [5] is designed to minimize communication cost in terms of the number of packet transmissions for better channel utilization. SADV [6] is a stable forwarding structure in road networks. It is based on relay nodes and reduces deviation in the delivery delay. These three schemes are for the multihop V2I data delivery, and the packet destination is a static node. Also, they utilize only vehicular traffic statistics to 1) estimate a link delay that is the delivery delay for a packet to be forwarded or carried over a road segment and 2) estimate a forwarding metric of end-to-end (E2E) delivery delay. Thus, these vehicular traffic statistics are macro-scoped vehicular information that describes the overall patterns of vehicle mobility in road networks.

Besides the forwarding schemes based on macroscoped vehicular information, the following three data-forwarding schemes have been proposed: TBD [1], TSF [2], and TMA [3]. These are based on microscoped vehicular information, such as vehicle trajectory. Based on vehicle trajectory information, TBD, TSF, and TMA are designed for the multihop V2I, I2V, and V2V data delivery, respectively. In this paper, we show how useful vehicle trajectory is in the design of data-forwarding schemes for vehicular networks. Because TMA [3] is an extension of TSF [2] for multicasting in vehicular networks, we focus on TSF along with TBD in this paper. Thus, the main ideas of TBD and TSF will be discussed to provide insight into the design of data-forwarding schemes with vehicle trajectory.

Machine-to-machine (M2M) communications have recently received a lot of attention within the networking community [22]. In a road network setting, M2M needs to allow drivers, passengers, and pedestrians to communicate with vehicles, infrastructure nodes, and Internet servers. This M2M is very important to realize vehicular cloud services that have been identified for next-generation vehicles [10]. Nowadays, most vehicles have more than 50 embedded computer components [11] including OBD systems. When vehicles connect to vehicular cloud via the infrastructure nodes, they can access the following vehicular cloud services: 1) automatic update of software related to systems embedded in the vehicle, 2) intelligent navigation for congested road networks, 3) automatic vehicle control to mitigate the damage in a road accident, 4) accident avoidance to prevent road accidents, and 5) the remote control of vehicles through mobile devices (e.g., smartphones and tablets). With these vehicular cloud services, DSRC-based data forwarding schemes provide vehicles with the network connectivity to the vehicular cloud through VANET at a lower cost by minimizing the usage of cellular networks such as 4G-LTE [23].

3 Delay Modeling

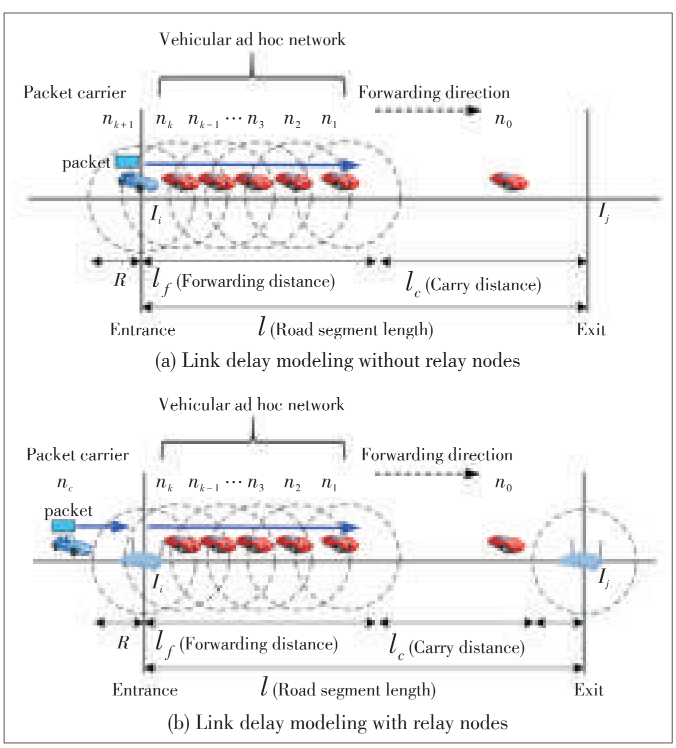
In this section, we describe link delay, E2E packet delivery delay, and E2E vehicle travel delay. We assume that vehicular traffic is one-way traffic to simplify delay modeling. Link delay modeling based on two-way traffic is not covered here.

3.1 Link Delay

In this subsection, we define link delay as the delay of a packet to be delivered over a road segment from its entrance intersection to its exit intersection using forward-and-carry. We consider link delay in the following two cases: 1) No relay node exists at each intersection, and 2) A relay node exists at each intersection as a temporary packet holder.

3.1.1 Link Delay for a Road Segment without Relay Nodes

We model link delay for a road segment without relay nodes at its intersections that are the end-points of the road segment. As shown in **Fig. 1a**, Packet Carrier n_{k+1} arrives at the entrance of road segment (I_i, I_j). The link delay over the road segment length l is the sum of the communication delay over the forwarding distance l_f and the carry delay over the carry distance l_c. For simplicity, we represent the link delay as the carry delay because the forwarding delay in milliseconds is negligible compared with the carry delay in seconds. That is, the carry delay is the dominant factor in the link delay.



▲**Figure 1. Link delay modeling for road segment.**

To compute the link delay, we first need to compute the forwarding distance l_f over road segment l and then compute the carry distance l_c as $l - l_f$. Let v be the average vehicle speed over the road segment. The road segment (I_i,I_j), the link delay d_{ij} can be computed as the carry delay as follows:

$$d_{ij} = \frac{l_c}{v} = \frac{l - l_f}{v}. \tag{1}$$

The expected link delay $E[d_{ij}]$ is computed as follows:

$$E[d_{ij}] = E\left[\frac{l_c}{v}\right] = E\left[\frac{l - l_f}{v}\right] = \frac{l}{v} - \frac{E[l_f]}{v}. \tag{2}$$

Thus, for $E[d_{ij}]$ in (2), the expected forwarding distance $E[l_f]$ needs to be computed. As shown in Fig. 1a, $E[l_f]$ can be computed as the sum of vehicle interdistances D_h for $h = 1 \ldots k$ from the entrance intersection I_i, leading to the connected vehicular ad hoc network. We assume that the vehicles arrive at the entrance intersection I_i of road segment (I_i, I_j) by the Poisson process of the arrival rate λ. In light-traffic vehicular networks that are our target settings, this assumption is validated from traffic measurements [24]. $E[l_f]$ is computed as the conditional expectation of the length of the connected vehicular ad hoc network, consisting of vehicle interdistances D_h (for $h = 1 \ldots k$) interconnected by the communication range R. The vehicle interdistance D_h is the product of vehicle interarrival time T_h and average vehicle speed v that is, $D_h = vT_h$. In [1], the expected forwarding distance $E[l_f]$ is computed as follows:

$$E\left[l_f\right] = E\left[D_h | D_h \leqslant R\right] \times \frac{P\left[D_h \leqslant R\right]}{P\left[D_h > R\right]}. \tag{3}$$

In (3), $E[l_f]$ is the product of 1) the average interdistance, denoted $E\left[D_h | D_h \leqslant R\right]$, of two consecutive vehicles within the same connected vehicular ad hoc network, and 2) the ratio of the probability, denoted $P\left[D_h \leqslant R\right]$, that the interdistance D_h is less than or equal to the communication range R to the probability, denoted $P\left[D_h > R\right]$, that the interdistance D_h is greater than the communication range R.

3.1.2 Link Delay for Road Segment with Relay Nodes

We model link delay for a road segment with relay nodes at its intersections. These nodes are end-points of the road segment. A relay node is placed at each intersection as a temporary packet holder for reliable I2V data delivery [2]. **Fig. 1b** shows link-delay modeling for a road segment (I_i, I_j) with relay nodes at its intersections I_i and I_j. For the case with relay nodes, we consider the following two cases: 1) immediate forwarding and 2) wait and carry. The first case is that packet carrier n_c forwards its packets to the head vehicle n_1 of the connected vehicular ad hoc network (comprising k vehicles from n_1 to n_k) via the relay node (denoted n_{k+1}) at the entrance I_i. The second case is that there are no vehicles within the communication range R of the entrance I_i moving toward exit I_j. In this case, packet carrier n_c forwards its packets to the relay node at entrance I_i, and the relay node holds the packets until a vehicle arrives at I_i and moves from I_i to I_j.

The link delay d for the two cases in Fig. 1b is given by

$$d = \begin{cases} \frac{l - l_f - R}{v} & \textit{for case } 1\text{: immediate forward,} \\ \frac{1}{\lambda} + \frac{l - R}{v} & \textit{for case } 2\text{: wait and carry.} \end{cases} \quad (4)$$

The expected link delay is computed as the conditional expectation of the link delay for the two cases as follows:

$$\begin{aligned} E[d] &= E[\text{link delay}|\text{forward}] \times P[\text{forward}] + \\ &\quad E[\text{link delay}|\text{wait}] \times P[\text{wait}] \\ &= \frac{l - R - E[l_f]}{v}\beta + \left(\frac{1}{\lambda} + \frac{l - R}{v}\right)(1 - \beta), \end{aligned} \quad (5)$$

where $P[\text{forward}] = \beta = 1 - e^{-\frac{\lambda R}{v}}$ and $P[\text{wait}] = 1 - \beta = e^{-\frac{\lambda R}{v}}$. The detailed derivation of $E[d]$ is given in [2, Appendix]. Similarly, the variance of the link delay is given by

$$Var[d] = E[d^2] - (E[d])^2, \quad (6)$$

where $E[d^2] = \frac{(l - R)^2 - 2(l - R)E[l_f] + E[l_f^2]}{v^2} \times \beta + \left(\frac{1}{\lambda} + \frac{l - R}{v}\right)^2 \times (1 - \beta)$ and $E[d]$ is (5).

The detailed derivation of $E[d^2]$ is given in [2, Appendix].

Finally, we can model the link delay as a Gamma distribution with mean $E[d]$ in (5) and variance $Var[d]$ in (6) because the link delay is a positive continuous random variable. Although we use this approximated distribution for the link delay, our forwarding design can accommodate any better distribution if available. Thus, the distribution of the link delay d_i for the directed edge $e_i \in E(G)$ in the road network graph G is $d_i \sim \Gamma(\kappa_i, \theta_i)$ such that $\theta_i = \frac{Var[d_i]}{E[d_i]}$ and $\kappa_i = \frac{E[d_i]}{\theta_i}$. Refer to [25] for the detailed the derivation of the parameters θ_i and κ_i. So far, the link delay over a road segment with relay nodes has been modeled. In next subsection, with this link delay, we will model E2E packet delivery delay.

3.2 E2E Packet Delivery Delay

We define E2E packet delivery delay as the packet delivery delay along a forwarding path from a source position to a destination position in the road network. We model this delay as the sum of the link delays of the road segments on the forwarding path. As in section 3.1.2, the E2E packet delivery delay, denoted P, can be modeled as a Gamma distribution with the mean and variance of the E2E packet delivery delay as follows, assuming that the forwarding path consists of n edges:

$$E[P] = \sum_{i=1}^{n} E[d_i]. \quad (7)$$

$$Var[P] = \sum_{i=1}^{n} Var[d_i]. \quad (8)$$

With the mean in (7) and the variance in (8), the E2E packet delay distribution can be modeled as $P \sim \Gamma(\kappa_p, \theta_p)$ such that $\theta_p = \frac{Var[P]}{E[P]}$ and $\kappa_p = \frac{E[P]}{\theta_p}$, as derived in [25].

3.3 E2E Vehicle Travel Delay

We define E2E vehicle travel delay as the time taken for a vehicle to move from its current position to its future position along its vehicle trajectory, which is the navigation path in the road network provided by a GPS navigator. It is known that the travel delay for a road segment in a light-traffic road network follows a Gamma distribution [26]. Thus, for a road segment $e_i \in E(G)$, the travel delay distribution is $t_i \sim \Gamma(\kappa_i, \theta_i)$ such that $\theta_i = \frac{Var[t_i]}{E[t_i]}$ and $\kappa_i = \frac{E[t_i]}{\theta_i}$, as derived in [25]. Note that even for a heavy-traffic road network, our design can accommodate any appropriate distribution from a mathematical model or empirical measurement.

For E2E vehicle travel delay, we take the same approach with the E2E packet delivery delay in section 3.2. Assuming that the vehicle trajectory consists of n edges, we have the mean and variance of the E2E vehicle delay distribution, denoted V, as follows:

$$E[V] = \sum_{i=1}^{n} E[t_i]. \quad (9)$$

$$Var[V] = \sum_{i=1}^{n} Var[t_i]. \quad (10)$$

With the mean in (9) and the variance in (10), the E2E vehicle delay distribution can be modeled as $V \sim \Gamma(\kappa_v, \theta_v)$ such that $\theta_v = \frac{Var[V]}{E[V]}$ and $\kappa_v = \frac{E[V]}{\theta_v}$, as derived in [25].

In next section, we describe three data forwarding schemes, VADD [4], TBD [1], and TSF [2]. Also, we model the packet delivery delay and vehicle travel delay.

4 Data-Forwarding Schemes

VADD enables us to invent TBD and TSF with vehicle trajectory for the V2I data delivery and the I2V data delivery, respectively. First, we explain how VADD computes a forwarding metric called EDD with only vehicular traffic statistics, used to select a next-hop vehicle in the V2I data delivery. Second, we describe how TBD plugs in vehicle trajectory in the computation of a forwarding metric EDD for the V2I data delivery. Last, we explain how TSF works for the I2V data delivery with our target point selection algorithm using the distributions of the destination vehicle's trajectory.

4.1 Vehicle-Assisted Data Delivery for V2I Data Delivery (VADD)

VADD [4] is a data-forwarding scheme for V2I data delivery. It is based on vehicular traffic statistics, such as the vehicle arrival rate and average speed per road segment along with the digital roadmaps provided by GPS navigation systems [15]. VADD is explained at first because TBD [1], as one of vehicle trajectory-based forwarding schemes, enhances the stochastic model of VADD with individual vehicle trajectory.

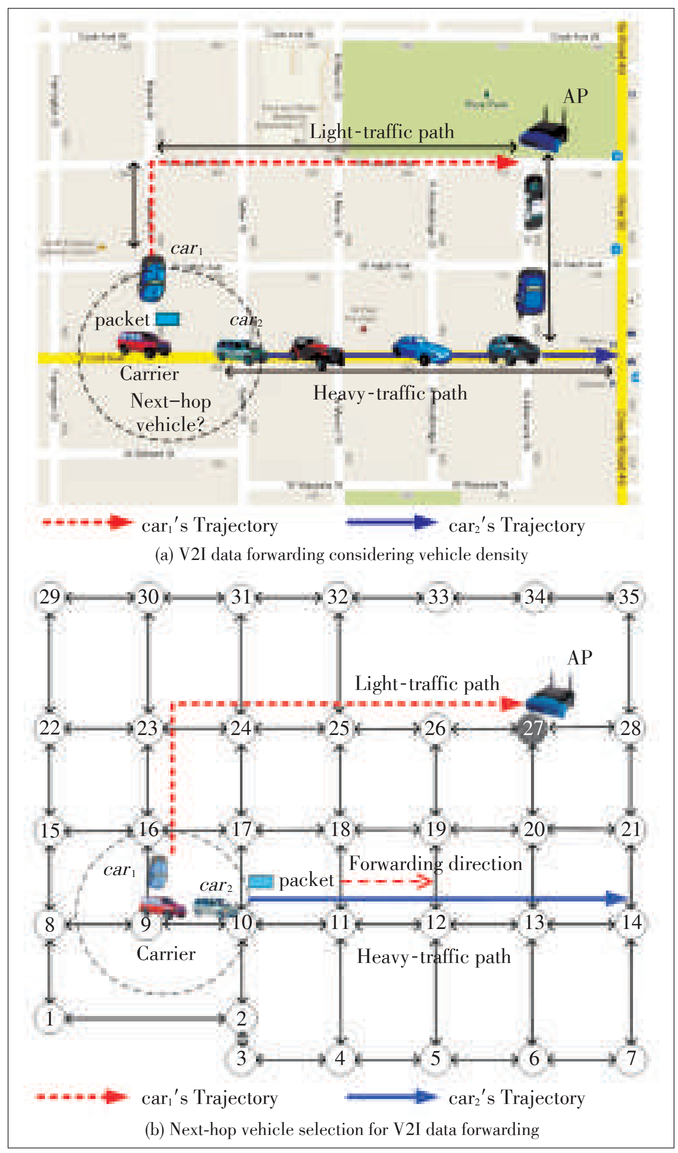
VADD aims to minimize delivery delay from vehicle to infrastructure node (e.g., AP). For example, the current packet carrier (denoted *carrier*) wants to deliver its packet to AP in the road network (**Fig. 2a**). *Carrier* has two neighboring vehicles, car_1 and car_2, within its communication range. The future trajectories of these cars are shown by solid or dotted arrows. Assume that the trajectory of car_1 passes through a light traffic path where a few vehicles are expected to move. On the other hand, the trajectory of car_2 passes through heavy traffic, and many vehicles are expected to move. Therefore, data forwarding over communication has a high chance using intermediate vehicles as packet forwarders during the packet's forward-and-carry process. In this case, definitely, Carrier needs to forward its packets to car_2 as a next-hop carrier rather than car_1, as shown in **Fig. 2b**. In VADD, to support this selection of a next-hop carrier based on vehicular traffic statistics, an EDD is computed as a forwarding metric by vehicles adjacent to the current packet carrier. A minimum-EDD vehicle will be selected as the next-hop carrier. Thus, the EDD computation is a key contribution in VADD.

Here, we explain how to compute EDD given the packet's destination (i.e., the location of the infrastructure node) along with the vehicular traffic statistics. Fig. 2b shows the road network graph as an abstract representation for the road network in Fig. 2a. This road network graph is a directed graph $G=(V,E)$, where V is the vertex set of intersections and E is the directed edge set of road segments. The EDD is computed on the basis of a stochastic model proposed by VADD [4]. Let d_{ij} be the expected link delay for edge e_{ij} in (2), discussed in section 3.1.1. Note that d_{ij} means $E[d_{ij}]$ in (2) for the simplicity of notation. Let D_{ij} be the EDD at the intersection i when a packet is delivered over the edge e_{ij}. The EDD D_{ij} is formulated recursively as follows:

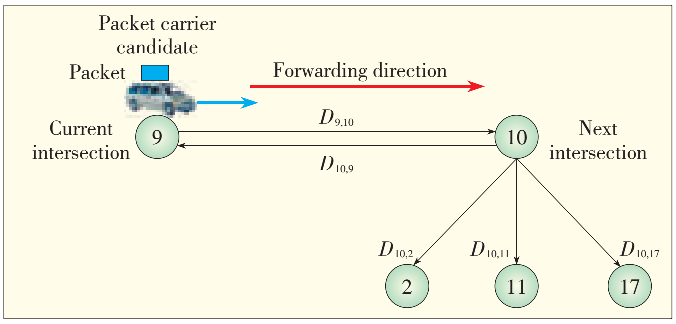
$$D_{ij} = d_{ij} + E[\text{delivery delay at } j \text{ by forward or carry}] = d_{ij} + \sum_{k \in N(j)} P_{jk} D_{jk}, \quad (11)$$

where $N(j)$ is the set of j's adjacent intersections. This recursive formation is reasonable because the packet delivered over edge e_{ij} arrives at intersection j and it is is forwarded to one of j's adjacent intersections, denoted k, with probability P_{jk} and the EDD D_{jk}. Refer to TBD in [1] for the detailed computation of the average forwarding probability P_{jk}.

Fig. 3 shows the EDD computation for edge $e_{9,10}$ where Packet Carrier Candidate is currently moving. The EDD $D_{9,10}$ is computed using (11) as follows: $D_{9,10} = d_{9,10} + P_{10,9}D_{10,9} + P_{10,2}D_{10,2} + P_{10,11}D_{10,11} + P_{10,17}D_{10,17}$. Even though VADD solves the data forwarding problem through the linear systems



▲**Figure 2. V2I data forwarding in road network.**



▲**Figure 3. EDD computation for Edge $e_{9,10}$.**

of recursive equations in (11), the limitation of VADD does not use the vehicle trajectory available for a better forwarding metric computation. In the next subsection, TBD [1] is used to take advantage of vehicle trajectory and improve VADD.

4.2 Trajectory-Based Data Forwarding for V2I Data Delivery (TBD)

TBD [1] is a data forwarding scheme to improve VADD for the V2I data delivery, using not only vehicular traffic statistics but also vehicle trajectory in the privacy-preserving manner. As an extreme example, assume that Fig. 2b describes the data forwarding in an extremely light-traffic vehicular network so that *carrier* has only car_1 and car_2 as the possible next-hop carriers in this road network. That is, we assume that only these three vehicles exist in the road network. The next-hop carrier candidates car_1 and car_2 are moving toward intersection 16 and intersection 10, respectively. One difference is that the trajectory of car_1 passes through AP, and the trajectory of car_2 is far away from the communication range of AP. In this case, car_1 should be selected by *carrier* as a next-hop carrier because car_1 will be able to deliver *carrier's* packets to AP with a shorter EDD than car_2. In this subsection, we explain how individual vehicles compute their EDD with their own trajectory in order to allow for this next-hop selection while they do not expose their own trajectory to other vehicles because of privacy concerns.

The main idea of TBD is to divide the data delivery process into the following two steps: 1) The packet carry process at the current carrier and 2) the delivery process after the packet leaves the current carrier. Note that in the case of light-traffic vehicular networks, a vehicle could carry a packet continuously over multiple edges along its trajectory until it meets a better next-hop carrier.

Suppose the current carrier has the trajectory T (i.e., a sequence of intersections to visit) as $T{:}1 \rightarrow 2 \rightarrow \cdots \rightarrow M$. Let C_{ij} be the total packet carry delay (i.e., travel delay) from intersection i to intersection j along the trajectory ($1 \leq i \leq j \leq M$). That is, C_{ij} is the sum of the carry delays of the road segments between intersections i and j such that $C_{ij} = \sum_{k=i}^{j-1} l_{k,k+1}/v$. The EDD for the trajectory T is given by

$$D = \sum_{j=1}^{M}\Big(P[\text{packet is carried from node 1 to } j] \times \big(C_{1j} + E[\text{delivery delay at } j]\big)\Big)$$
$$= \sum_{j=1}^{M}\left(\left(\prod_{h=1}^{j-1} P^{c}_{h,h+1}\right) \times \left(C_{1j} + \sum_{k \in N(j)} P'_{jk} D_{jk}\right)\right), \quad (12)$$

where P'_{jk} is the forwarding probability to forward a packet at intersection j to another vehicle moving toward intersection k (computed in (6) in [1]), $P^{c}_{h,h+1}$ is the carry probability to carry a packet from intersection h to $h+1$ such that $P^{c}_{h,h+1} = 1 - \prod_{k \in N(h)} P'_{h,h+1}$, and D_{jk} is the EDD at edge e_{jk} in (11).

For example, **Fig. 4** shows the EDD computation for a packet carrier candidate with the trajectory ($T{:}10 \rightarrow 11 \rightarrow 12$). The EDD D is computed by (12) as follows:

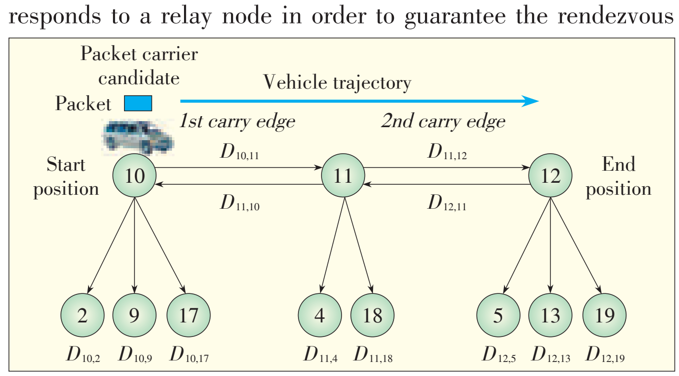
$$\begin{aligned} D = {} & P'_{10,2}D_{10,2} + P'_{10,9}D_{10,9} + P'_{10,11}D_{10,11} + P'_{10,17}D_{10,17} + \\ & P^{c}_{10,11}(C_{10,11} + P'_{11,4}D_{11,4} + P'_{11,10}D_{11,10} + P'_{11,12}D_{11,12} + P'_{11,18}D_{11,18}) + \\ & P^{c}_{10,11}P^{c}_{11,12}(C_{10,12} + P'_{12,5}D_{12,5} + P'_{12,11}D_{12,11} + P'_{12,13}D_{12,13} + P'_{12,19}D_{12,19}). \end{aligned} \quad (13)$$

Therefore, TBD allows individual vehicles to calculate their own EDD based on their own trajectory so that the packet carrier can select the best next-hop carrier among its neighboring vehicles. However, TBD is designed for the static packet destination. Thus, when the destination is moving in the I2V data delivery, we need a totally different approach that takes into account the mobility of the destination vehicle. In the next subsection, we introduce TSF [2] for multihop I2V data delivery.

4.3 Trajectory-Based Statistical Data Forwarding for I2V Data Delivery (TSF)

TSF [2] is a data-forwarding scheme for multihop I2V data delivery, which involves the packet destination vehicle trajectory. **Fig. 5** shows I2V data delivery from AP_1 to Destination Vehicle. TSF for I2V has one significant difference from VADD and TBD for V2I in that TSF requires relay nodes at intersections as temporary packet holders that are not directly connected to the wired network for the deployment cost reduction. The relay nodes are required for the reliable I2V data delivery from AP to a destination vehicle so that the delivery delay standard deviation is bounded to deliver packets from AP to the moving destination vehicle in a timely manner [2], [6].

The challenge for I2V is in selecting a target point that corresponds to a relay node in order to guarantee the rendezvous



▲**Figure 4. EDD computation for vehicle trajectory.**

of the packet from AP and the moving destination vehicle. In Fig. 5, AP_1 selects intersection 13, denoted n_{13}, as a target point through the current position and trajectory of Destination Vehicle. The current positions and trajectories of vehicles are available to APs via TCC [18] because the vehicles regularly report their current position and trajectory to TCC for the location management in TCC for the mobile vehicles like in Mobile IPv6 [27]. Thus, TCC is a home agent in managing the location of vehicles in the similar way with Mobile IPv6 so that APs can get the estimated current position and vehicle trajectory of a destination vehicle from TCC.

In TSF, the target point selection is performed with the following two delay distributions: 1) Vehicle delay distribution from Destination Vehicle's current position to a Target Point and 2) Packet delay distribution from AP to a Target Point. **Fig. 6** shows the packet delay distribution from AP_1 to target point candidate n_{13} and the vehicle delay distribution from Destination Vehicle's current position n_{10} to target point candidate n_{13}. For each intersection as a target point candidate along Destination Vehicle's trajectory, we can draw a pair of delay distributions, as in Fig. 6.

To optimize delivery, we formulate the target point selection as follows. Let I be a set of intersections along Destination Vehicle's trajectory. Let P_i be the packet delay from AP to target point candidate i. Let V_i be the vehicle delay from Destination Vehicle's current position to target point candidate i. As a target point, TSF selects an intersection to minimize the packet delivery from AP to Destination Vehicle, while satisfying the user-defined delivery probability threshold α (e.g., 95%) as follows:

$$i^* \leftarrow \arg\min_{i \in I} E[V_i] \text{ subject to } P[P_i \le V_i] \ge \alpha. \quad (14)$$

In (14), $P[P_i \le V_i]$ is the delivery probability that the packet will arrive at intersection i earlier than Destination Vehicle. In (14), $E[V_i]$ is the actual packet delivery delay from AP to Destination Vehicle. This is because the packet held by the relay node at intersection i is forwarded to Destination Vehicle when Destination Vehicle passes through intersection i after $E[V_i]$.

We model the packet delay distribution and the vehicle delay distribution in Fig. 6 as Gamma distributions so that $P \sim \Gamma(\kappa_p, \theta_p)$ and $V \sim \Gamma(\kappa_v, \theta_v)$. These are discussed in section 3.2 and section 3.3, respectively. If more accurate delay distributions are available, our TSF design can accommodate those better distributions for a better target point selection.

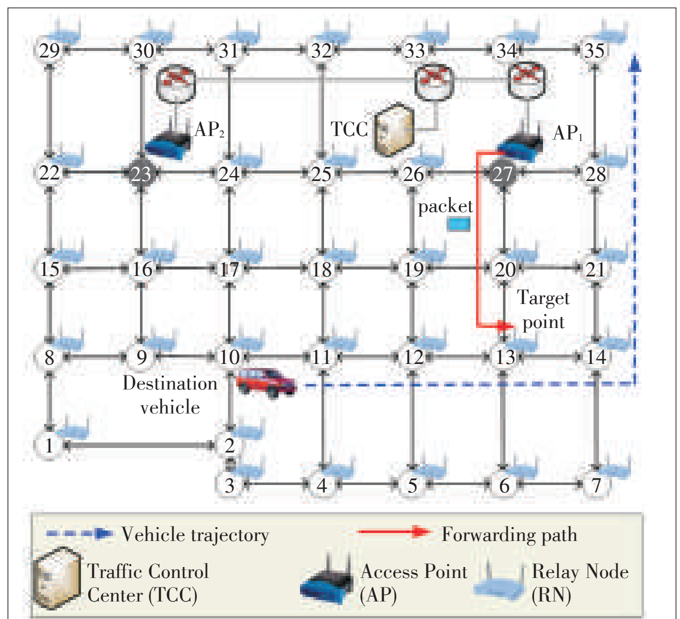
Given the packet delay distribution and the vehicle delay distribution, the delivery probability $P[P_i \le V_i]$ is given by

$$P[P_i \le V_i] = \int_0^{TTL} \int_0^{v} f(p) g(v) dp dv, \quad (15)$$

where $f(p)$ is the probability density function (PDF) of packet delay p, $g(v)$ is the truncated PDF of vehicle delay v with the integration upper bound TTL that is the packet's Time-To-Live (TTL). Note that since the packet is discarded after TTL, the portion of the delivery probability for vehicle delay v becomes zero after TTL.

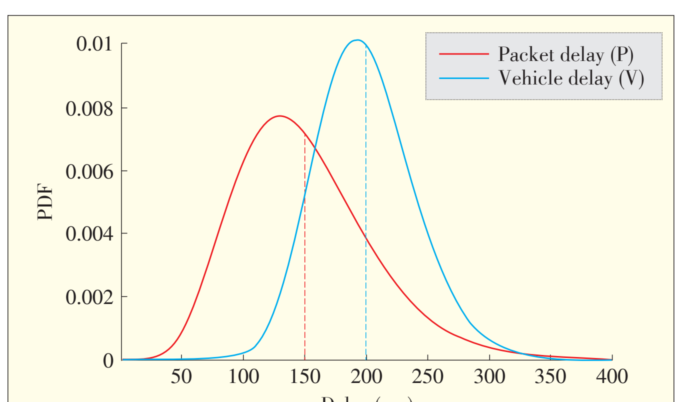
TSF can be used for the multihop V2V data delivery in the combination of V2I and I2V. That is, Source Vehicle sends a packet to a nearby AP using TSF (or TBD) for V2I data delivery. Source Vehicle regards AP's intersection as a target point (destination). The AP contacts TCC to locate Destination Vehicle and obtains the corresponding trajectory to compute a target point. With the target point, AP sends the packet toward the target point for I2V data delivery to Destination Vehicle.

TSF can be extended to support multicast from AP to a multicast group vehicles moving in a road network. As a multicast version of TSF, we propose TMA [3]. TMA computes the multiple target points of multicast group vehicles in the same way



PDF: probability density function

▲Figure 5. I2V data forwarding in road network.



▲Figure 6. Packet delay distribution and vehicle delay distribution.

that TSF does. With these multiple target points, TMA constructs a minimum Steiner Tree for multicast data delivery so that multicast delivery cost can be minimized and multicast data can be more efficiently shared between vehicles in a multicast group.

One limitation of TSF is that relay nodes need to be deployed as infrastructure nodes for reliable I2V data delivery. In future work, we will develop a data-forwarding scheme that supports both I2V and V2V data delivery without relay nodes and fully utilize the trajectories of vehicles moving in a target road network. In the next section, we analyze three forwarding schemes discussed in this section.

5 Analysis of Forwarding Schemes

Table 1 shows a comparison of the VANET data-forwarding schemes VADD, TBD and TSF. VADD and TBD only support V2I, and their target application is road condition reports. TSF supports V2I, I2V and V2V, which means there are more target applications, such as road condition sharing and cloud services (e.g., navigation and location-based services). These three forwarding schemes use vehicular traffic statistics for forwarding-metric computation. Except for VADD, the other two schemes TBD and TSF take advantage of vehicle trajectory for more efficient forwarding metric computation. TSF supports the more forwarding types, such as V2I, I2V, and V2V.

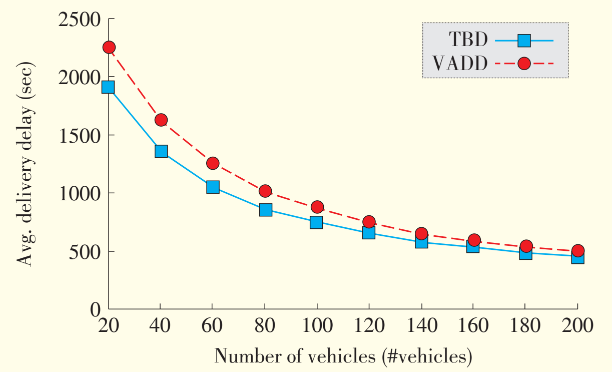
All three forwarding schemes require access points for connectivity to a wired network, such as the Internet. TSF additionally requires relay nodes and traffic control center for reliable multihop I2V (or V2V) data delivery, and protects privacy by not exposing the vehicle trajectories. Thus, for vehicular cloud services through vehicular networks, TSF is recommended because it supports bi-directional data communications between vehicles and infrastructure nodes (e.g., AP).

In the simulations, we evaluated the performance of VADD, TBD, and TSF in an 8.25 km × 9 km road network with 49 intersections. The DSRC communication range is 200 m. The vehicles move in the road network according to a Hybrid Mobility model of City Section Mobility model [28] and Manhattan Mobility model [29]. The simulation configuration can be found in the performance evaluation of TBD [1] and TSF [2].

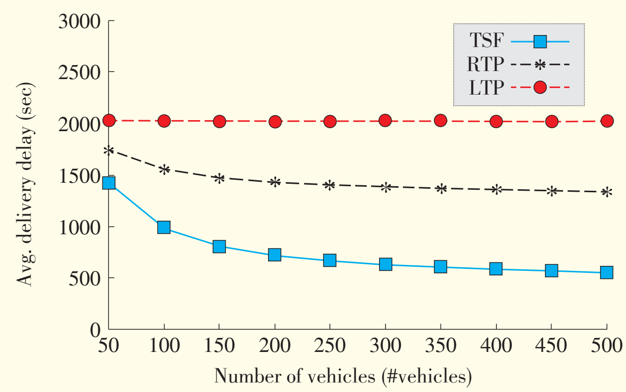
Fig. 7 shows the performance of VANET data-forwarding schemes. For multihop V2I data delivery, Fig. 7a shows the performance of TBD and VADD in average delivery delay by the number of vehicles (i.e., vehicular density) [1]. TBD has a shorter delivery delay than VADD from the lowest vehicular density to the highest vehicular density by a more effective delivery delay estimation using the individual vehicle trajectory. This indicates that TBD provides better V2I data delivery than VADD. For multihop I2V data delivery, Fig. 7b shows TSF, Random Target Point (RTP), and Last Target Point (LTP) [2]. These are different in terms of the target point selection mechanism for a rendezvous point of the packet and destination vehicle. RTP selects a target point as a random intersection among the intersections along the destination vehicle's trajectory. LTP selects a target point as the last intersection of the destination vehicle's trajectory. On the other hand, TSF selects a target point by the optimization in (13) with the packet delay distribution and vehicle delay distribution shown in Fig. 6. TSF has a shorter delivery delay than both RTP and LTP by the optimal target point selection (Fig. 7b). Therefore, the vehicle trajectory is very important information in the design of the data forwarding schemes for either V2I or I2V data delivery.

▼Table 1. The comparison among VANET data forwarding schemes

Scheme	Type	Vehicular Statistics	Vehicle Trajectory	Infrastructure Nodes	Privacy Exposure	Target Application
VADD	V2I	Yes	No	Access points	No	Road condition report
TBD	V2I	Yes	Yes	Access points	No	Road condition report
TSF	V2I, I2V, V2V	Yes	Yes	Access points, relay nodes, traffic control center	No	Road condition sharing, cloud services (e.g., navigation and location-based services)



(a) TBD and VADD comparison for delivery delay by vehicle number for V2I



(b) LTP, RTP, and TSF comparison for delivery delay by vehicle number for I2V

▲**Figure 7. The performance evaluation of VANET data forwarding schemes.**

6 Conclusions

In this paper, we have described TBD and TSF data-forwarding schemes based on vehicle trajectory in vehicular networks. The vehicle trajectory is a good asset in the design of data-forwarding schemes for multihop V2I or I2V data delivery because it allows for either better forwarding metric computation or better estimation of the location of the packet destination vehicle. In future work, we will investigate more of the characteristics of vehicle trajectory in order to achieve better data forwarding performance, considering the minimization of trajectory sharing overhead and the privacy protection on trajectory. In particular, we will design and implement a new data-forwarding scheme to support multihop V2I, I2V, and V2V data delivery without any relay nodes to reduce deployment cost. For this new data-forwarding scheme, we will investigate how to fully utilize the trajectories of vehicles moving in a target road network. That is, this data forwarding scheme will investigate how to combine packet carrying process and packet forwarding process by predicting the encounter sequence of vehicles as the forwarding chances between the current packet carrier and next-packet carrier candidates with vehicle trajectories.

References

[1] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based data forwarding for light-traffic vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743-757, May 2011. doi: 10.1109/TPDS.2010.103.

[2] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp 1523-1537, Oct. 2012. doi: 10.1109/TMC.2011.189.

[3] J. Jeong, T. He, and D. Du, "TMA: trajectory-based multi-anycast forwarding for efficient multicast data delivery in vehicular networks", *Elsevier Computer Networks*, vol. 57, no. 13, pp. 2549-2563, Sept. 2013. doi: 10.1016/j.comnet.2013.05.002.

[4] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, May 2008. doi: 10.1109/TVT.2007.901869.

[5] A. Skordylis and N. Trigoni, "Delay-bounded routing in vehicular ad hoc networks," in *ACM MOBIHOC*, Hong Kong, China, May 2008, pp. 341–350. doi: 10.1145/ 1374618.1374664.

[6] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proc. ACM Int' l Workshop Vehicular Ad Hoc Networks*, Montreal, Canada, Sept. 2007, pp. 59-68. doi: 10.1145/ 1287748.1287758.

[7] G. Dimitrakopoulos, "Intelligent transportation systems," *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 77–84, Mar. 2010.

[8] OECD report in 2008. (2010, Aug. 26). *South Korea's Road Fatality Rate Among Worst* [Online]. Available: http://english.yonhapnews.co.kr/national/2010/08/26/89/0302000 000AEN20100826005300315F.HTML

[9] M. Ferreira, R. Fernandes, H. Conceicao, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," *ACM VANET*, Chicago, IL, USA, Sept. 2010. pp. 85–90. doi: 10.1145/1860058.1860077.

[10] M. Aoyama, "Computing for the next-generation automobile," *IEEE Computer*, vol. 45, no. 6, pp. 32–37, Jun. 2012. doi: 10.1109/MC.2012.153.

[11] *On-Board Diagnostics (OBD)* [Online]. Available: http://epa.gov/otaq/regs/im/obd/ index.htm

[12] Y. Morgan, "Notes on DSRC & WAVE standards suite: its architecture, design, and characteristics," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 504–518, Oct. 2010. doi: 10.1109/SURV.2010.033010.00024.

[13] ETSI. *DSRC Standardization* [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/intelligent-transport/dsrc

[14] SAE International. (2009). *DSRC Implementation Guide* [Online]. Available: http://www.sae.org/standardsdev/dsrc/DSRCImplementationGuide.pdf

[15] Garmin Ltd. (2012). *Garmin Traffic* [Online]. Available: http://www8.garmin.com/ traffic, 2012.

[16] Smartphone Navigator. (2011, Jan. 14). *Apps with Maps: 11 iPhone GPS Apps Compared* [Online]. Available: http://www.macworld.com/article/1156720/gps.html

[17] H. Yomogita. *Mobile GPS Accelerates Chip Development* [Online]. Available: http://techon.nikkeibp.co. jp/article/HONSHI/20070424/131605/

[18] Philadelphia Department of Transportation. *Traffic Control Center* [Online]. Available: http://philadelphia.pahighways.com/philadelphiatcc.html

[19] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, Jan. 2011. doi: 10.1109/JSAC.2011.110113.

[20] J. Jeong, "Vehicle trajectory-based forwarding schemes for vehicular ad hoc networks," *KICS Information and Communications Magazine*, vol. 29, no. 8, Aug. 2012.

[21] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, Apr. 1999. doi: 10.1109/98.760423.

[22] *Machine to Machine (M2M) System Requirements Document (SRD)*, IEEE 802.16p-10/0004, 2011.

[23] C. D. Monfreid, "The LTE network architecture — a comprehensive tutorial," Tech. Rep., 2009.

[24] N. Wisitpongphan, F. Bai, P. Mudalige, and O.K. Tonguz, "On the routing problem in disconnected vehicular ad hoc networks," *IEEE INFOCOM*, Anchorage, AK, USA, May 2007, pp. 2291–2295. doi: 10.1109/INFCOM.2007.267.

[25] M. DeGroot and M. Schervish, *Probability and Statistics*, third ed. Boston, MA: Addison-Wesley, 2001.

[26] D. S. Berry and D. M. Belmont, "Distribution of vehicle speeds and travel times," *Proc. Second Berkeley Symp. Math. Statist. and Prob.*, Berkeley, CA: Univ. of Calif. Press, 1951, pp. 589–602..

[27] C. Perkins, D. Johnson, and J. Arkko, *Mobility Support in IPv6*, IETF RFC 6275, July 2011.

[28] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Comm. and Mobility Computing, Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, Aug. 2002. doi: 10.1.1.13.268.

[29] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad-hoc networks," *IEEE INFOCOM*, San Francisco, CA, USA, 2003, vol. 2, pp. 825–835. doi: 10.1109/ INFCOM.2003.1208920.

Manuscript received: February 20, 2014

Biographies

Jaehoon (Paul) Jeong (pauljeong@skku.edu) is an assistant professor in the Department of Software, Sungkyunkwan University, Republic of Korea. He received his PhD degree in computer science and engineering from University of Minnesota, USA, in 2009. He received his BS degree in information engineering from Sungkyunkwan University, Republic of Korea, and his MS degree in computer engineering from Seoul National University, Republic of Korea. His research interests include vehicular networks, cyber-physical systems, and navigation systems.

Tian He (tianhe@cs.umn.edu) is an associate professor in the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, USA. He received his PhD degree under Professor John A. Stankovic at the University of Virginia, USA, in 2004. He has authored or co-authored more than 90 papers in premier sensor network journals and conferences and has more than 4000 citations. His publications have been selected as graduate-level course materials by over 50 universities in the United States and other countries.

David H. C. Du (du@cs.umn.edu) is the Qwest chair professor in the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, USA. He received his BS degree in mathematics from National Tsing-Hua University, Taiwan. He received his MS and PhD degrees in computer science from the University of Washington, USA in 1980 and 1981. His research interests include cyber security, sensor networks, multimedia computing, storage systems, and high-speed networking. He is a fellow of the IEEE.

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 004
http://www.cnki.net/kcms/detail/34.1294.TN.20140306.1003.001.html, published online March 6, 2014

Unveiling the Challenges in Improving Data Availability in Vehicular Networks with Network Coding

Zhenni Feng[1], Yanmin Zhu[1,2], Qian Zhang[3], and Min Gao[4]
(1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;
2. Shanghai Key Lab of Scalable Computing and Systems, Shanghai 200240, China;
3. Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China;
4. HKUST Fok Ying Tung Graduate School, Guangzhou 511400, China)

Abstract
Retrieving data from mobile source vehicles is a crucial routine operation for a wide spectrum of vehicular network applications, including road surface monitoring and sharing. Network coding has been widely exploited and is an effective technique for diffusing information over a network. The use of network coding to improve data availability in vehicular networks is explored in this paper. With random linear network codes, simple replication is avoided, and instead, a node forwards a coded block that is a random combination of all data received by the node. We use a network-coding-based approach to improve data availability in vehicular networks. To determine the feasibility of this approach, we conducted an empirical study with extensive simulations based on two real vehicular GPS traces, both of which contain records from thousands of vehicles over more than a year. We observed that, despite significant improvement in data availability, there is a serious issue with linear correlation between the received codes. This reduces the data-retrieval success rate. By analyzing the real vehicular traces, we discovered that there is a strong community structure within a real vehicular network. We verify that such a structure contributes to the issue of linear dependence. Then, we point out opportunities to improve the network-coding-based approach by developing community-aware code-distribution techniques.

Keywords
vehicular networks; network coding; data availability

1 Introduction

Vehicles with radio technology specially designed for inter-vehicle communications, such as dedicated short range communication (DSRC), form vehicle networks [1]. Because they are low-density and fast-moving, vehicular networks are usually sparse, rely on delay-tolerant networking, and leverage the "carry-and-forward" paradigm for data delivery [2]. A node can forward data as it contacts another node.

A wide range of monitoring applications [3], [4] can be built in a delay-tolerant vehicular network. In such a network, source vehicles collect sensing data as they move and share this data with other vehicles or users. In road surface monitoring and sharing [5], a vehicle periodically detects the surface conditions of roads on which it is travelling. Then, when road surface information is needed, a vehicle can retrieve this information that has been detected by other vehicles. Such retrieval is also needed for other urban-sensing applications.

An intuitive method for realizing the monitoring application based on vehicular networks is to use 3G/GPRS cellular data channels. All sensing data can be transferred to a central server that then publishes or pushes aggregated data back to querying nodes. However, this method has two drawbacks. First, introducing a central server increases the overall cost and complexity of realizing such a monitoring application. It also leads to the potential problem of single-point failure. Second, the use of cellular data channels incurs cost, especially when there are multiple types of sensor data. These applications rely on the voluntary participation of contributing vehicles. It is not reasonable to require the volunteer vehicles to pay for the incurred cost of data uploading.

Therefore, this paper explores a new approach to sharing such sensor data between vehicles. A querying node retrieves the desired sensing data directly from vehicles in the network, not through a central server. The key to improving data-retrieval is to increase data availability so that a retrieving vehicle can easily acquire desired data from vehicles that it contacts. An extreme approach is for every source node to replicate its

This work is supported by China 973 Program (2014CB340303), NSFC (No. 61170238, 60903190), and National 863 Program (2013AA01A601).

data on every other node in the network, and a retrieving node can easily acquire the data of a given node by retrieving it from any node that the node contacts.

However, such blind replication introduces high communication overhead, which is impractical in a real vehicular network. Contact opportunities can be scarce in a real-world vehicular network. According to [6], the average pair-wise inter-contact time in a vehicular network with 2000 taxis can be as long as two hours.

There are several existing approaches to improving data availability [7]–[9]. The basic idea of these approaches is to replicate data items to other nodes. Challenged by infrequent and intermittent encounters, these replication-based schemes are prone to the coupon collector's problem [10]; that is, a node may repeatedly receive replicas of data items that it has already had but fails to obtain some specific items.

There are two problems with the existing replication approaches. First, high communication cost can be incurred by distributing replicas of data items to other vehicles. This may be acceptable in dense ad hoc networks [11] or wired P2P networks [12], but it is inappropriate in vehicular networks. Second, it takes a retrieving node a long time to obtain all the desired data items.

Network coding [13] has been widely studied as a way of improving network throughputs [14], [15]. Recently, it has also been recognized as an effective technique for diffusing information over a network [16]. In this paper, we explore the use of network coding to improve data availability in vehicular networks. By using random linear network codes, simple replication is avoided, and instead, a vehicle forwards a coded block that is a random combination of all data received by the vehicle. In this way, the data blocks maintained by different vehicles are different. Suppose there are m data items. Ideally, a vehicle can recover all these items if it can retrieve m coded blocks from the vehicles it contacts. In this way, replicas are not repeatedly retrieved, and the process of successfully retrieving m items is accelerated.

In this paper, we use a network-coding-based approach to improving data availability in vehicular networks. To determine the feasibility of this approach, we conducted an extensive empirical study using simulations based on a large data set of real vehicular GPS traces collected from approximately 2600 taxis in Shanghai. We observed the following: First, data availability and data retrieval are improved compared with existing replication-based approaches. Second, the set of coded blocks retrieved by a vehicle is highly correlated, which results in dramatic reduction in data availability because a larger number of coded blocks need to be retrieved to recover m items. This significantly deviates from our intuition that the received blocks should be linearly uncorrelated given the nature of random linear network codes.

To explain the problem of linear correlation with coded blocks, we delved into the characteristics of vehicular networks and analyzed the large data set of real traces. We found that there is strong heterogeneity with contact patterns with vehicles. This means that some vehicles may contact other vehicles frequently, and some vehicles may rarely contact other vehicles. Furthermore, we discovered that there is a clear community structure within a vehicular network. By using the community-detection algorithm [17] on a 1000-node vehicular network comprising the taxis from the real traces, the whole network can be divided into 33 communities. Vehicles within a community contact each other many more times than vehicles from different communities. This phenomenon may cause the problem of linear correlation of coded blocks. With random linear network codes, coded blocks in the same community tend to be correlated.

To validate our hypothesis, we introduce two synthetic vehicular networks: one with six communities and one without clear communities. By simulating the two synthetic networks, we verify that a network with communities suffers from linear correlation; however, a network without communities is free from such problem. We confirm the issue with network coding for data availability in vehicular networks and point out the opportunities for alleviating such an issue by taking community structures of the vehicular network into account.

2 Network-Coding-Based Approach

2.1 Preliminaries

We consider an n-node vehicular network that may distribute over a vast area and where network connectivity is unavailable most of the time. We call such a vehicular network a sparse vehicular network. There are no access points (APs) or roadside infrastructure. Vehicles in the network are denoted $V=\{v_1,v_2,...,v_n\}$. A node can communicate with other nodes within the transmission range r of the node. In such a sparse vehicular network, a vehicle can forward data when it comes in contact with another vehicle. The duration of the contact is usually short, and only a limited number of packets can be transmitted.

Among all the vehicles, there is a subset M of m source vehicles, each of which may generate data periodically so that $M \subset V$ and $m \leqslant n$. The length of the period is denoted T. At the beginning of each period, each source vehicle generates one data item. Data items are of equal size and different from each other. We consider a situation where the data does not change very quickly, and it is practical to take some time to prepare for retrieving. There is a single stream in the network, and it is easy to extend to multiple streams when the data in each stream is transmitted separately. If there are multiple streams, a vehicle can randomly select a coded block from the available streams on the vehicle and transfer it to the encountered vehicle. Because there are more streams, the opportunity for a specific stream to distribute across the network decreases.

Each data item has a unique identification in the vehicular network. Each source node randomly selects a number from a large range as the identification of the data item it generates. Thus, each data item has a unique id when the range is enough large.

A retrieving node seeks to retrieve all data items generated by the source nodes in a specific period. Such retrieval should occur within a given time constraint T_R. Data items retrieved beyond the time constraint are considered meaningless. Note that $T > T_R$.

2.2 Design

The goal of improving data availability in vehicular networks is to allow a retrieving node to retrieve as many data items as possible within the given retrieval time constraint. In the meantime, the startup time for distributing data replicas or coded blocks should be minimized in order to improve the responsiveness of the whole system. Importantly, the communication cost of data distribution should be reduced because of the scarcity of communication opportunities in the vehicular network.

We next present the design of a network-coding-based approach to improving data availability in vehicular networks. This approach has three main phases: code distribution, code retrieval, and data recovery (**Fig. 1**).

In phase one, the network distributes coded blocks that are encoded with original data items or other coded blocks. Specifically, when encountering another vehicle, each vehicle forms a coded block by taking a random linear combination of all coded blocks it has stored.

In addition to the newly coded block, other information, such as random coefficients and the set of ids of all original data items encoded by the newly coded block, is also transmitted along with the coded block.

To ensure system responsiveness, a time constraint T_D is imposed in the code-distribution phase. To avoid useless transmissions, the sender and receiver exchange information to judge whether the newly coded block is useful for the receiver. The sender sends information in the head of the coded block to the receiver before sending the whole block. The receiver judges whether the incoming coded block is linearly independent from coded blocks it already holds and sends the result back. Consequently, the sender continues to send the rest of the coded block if it is independent, and vice versa. To start the distribution, every source node initially generates a coded block that contains its own data item.

In phase two, a retrieving node v_k tries to retrieve as many coded blocks as possible. Specifically, when v_k encounters a vehicle u, it requests a coded block from u. Upon receiving such a request, u forwards a coded block to v_k by taking a random combination of all coded blocks it holds. This phase terminates after T_R has passed since v_k started the data-retrieval process.

In phase three, after v_k has retrieved a set of coded blocks, v_k tries to recover the original data items. The coefficients in the head of a coded block denote a linear equation, and all coded blocks that the retrieving node holds constitute a system of linear equations. The standard solution of Gaussian elimination can solve the system of linear equations. If there are more than m linearly independent equations, m original data items can be fully recovered. The length of phase three is considerably short. The typical decoding algorithm is based on solving linear equation system, and the typical method is Gaussian elimination. Thus, the phase three is much shorter than the other two phases.

3 Revealing the Linear Dependency of Codes

In this section, we determine the feasibility of the network-coding-based (NC) approach by running trace-driven simulations. We compare NC with a baseline approach, which is based on simple replication (SR). In SR, when a node v_a encounters node v_b, v_a randomly chooses a data item that node v_b does not have and sends it to v_b. This is different from what occurs in NC.

To simulate the vehicular network, we used a relatively simple disk model, i.e., two vehicles within the communication range can communicate with each other.

The simulations were driven by one large data set of real vehicular GPS traces from Shanghai taxis. The dataset was collected from approximately 2600 taxis in Shanghai. The trace of a taxi contains the real-time position (longitude and latitude) with a corresponding timestamp. The interval between two records varies from 15 s to several minutes. All the traces were

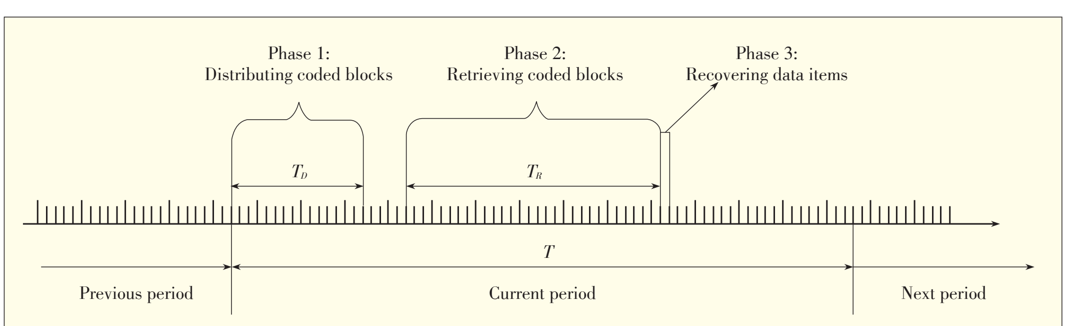


Figure 1.
Three phases in the network-coding-based approach for data retrieval.

recorded from January 2006 to December 2007.

Although data availability in a vehicular network is improved by using random linear network codes, there is a serious issue: coded blocks received by the retrieving node are linearly correlated. Even if m coded blocks are retrieved by the retrieving node, it is possible that all original data items may not be recovered because of the linear dependence of these coded blocks. In this subsection, we investigate the issue of linear dependence of received coded blocks.

We first define a metric called linear independence ratio γ to characterize the degree of linear independence of received coded blocks.

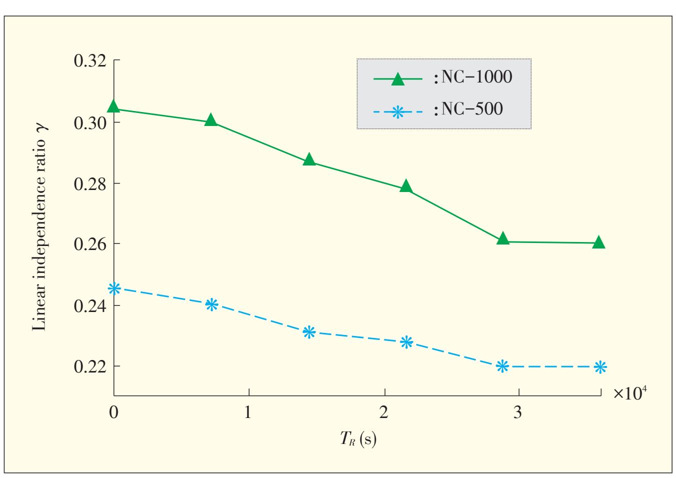
The linear independence ratio γ of a set of C coded blocks is the ratio of the maximum number of linearly independent codes to the total number of codes. If the corresponding coefficient matrix of C is $S(\mathrm{C})$, then the linear independence ratio is

$$\gamma(\mathrm{C}) = \frac{rank\,(S(\mathrm{C}))}{|\mathrm{C}|} \qquad (1)$$

where $|\mathrm{C}|$ is the size of set C, and $rank(S(\mathrm{C}))$ is the rank of matrix $S(\mathrm{C})$.

Fig. 2 shows γ of the set of the coded blocks that the retrieving node has received within the time constraint of codes retrieval. The size of the set of coded blocks is fixed if the set can be used to fully recover all m data items, and γ is much less than 1. When there are 1000 vehicles in the network and the time constraint for code retrieval is 1.5×10^4 s, the ratio is as low as 28%.

We expect that γ is close to one so that all data items can be quickly recovered when the number of received codes is close to m. However, the real γ experienced by retrieving nodes in a vehicular network is much smaller than 100%. This indicates that the retrieving node can hardly recover m data items, even though it receives m coded blocks, because of the linear correlation of the received codes. The phenomenon deviates from our intuition that random linear network codes would improve data availability in vehicular networks.



▲Figure 2. Linear independence ratio of received blocks vs. time constraint of codes retrieval.

4 Unveiling Community Structures

We suggest that the community structures in the network cause linear dependence, which is verified in section 5.

Topology of complex networks is a hot topic, and much attention has been given to large-scale networks, such as social networks, mobile networks and the internet, with millions or even billions of nodes [17], [18]. One of the most promising research approaches to these networks has been to divide the large network into several parts where inter-part connections are much sparser than intra-part connections, and a part forms a community. This area of research is called community detection.

Most popular community-detection algorithms are based on the modularity metric. Modularity is a common indicator of the quality of the detected community structures and is a scalar value ranging from −1 to 1. Given a graph G, the modularity Q is defined as

$$Q = \frac{1}{2m}\sum_{i,j}(A_{i,j} - \frac{k_i k_j}{2m})\delta\left(c_i, c_j\right), \qquad (2)$$

where $A_{i,j}$ is the weight of the edge between i and j; m is the number of edges; k_i is the degree of node v_i; c_i identifies the community that v_i belongs to; and function $\delta(i,j)$ is 1 if $i=j$ and 0 otherwise. A high modularity (typically higher than 0.3) means there are clear community structures. From (2), any pair of nodes belongs to different communities and contributes nothing to the value of modularity; that is, a graph with no edges has a modularity of zero.

We use the state-of-the-art algorithm proposed by Blondel et al., which is a simple, efficient algorithm for extracting community structures [17]. It is an agglomerative, bottom-up algorithm that initially assigns every vertex to a separate community and then merges communities until the overall modularity is maximized.

Given the contact records of every pair of nodes in a vehicular network, we construct such a graph using the following steps:

1) Construct a complete graph. Every vehicle is regarded as a node in the graph, and every pair of nodes shares an edge. Every edge has an attribution called number of contacts.
2) Compute the contact rate. Compute the number of contacts for every pair of nodes during a given time period in order to obtain the contact rate of every pair.
3) Delete edges. Select a proper threshold for the contact rate. If the value of the attribution of each pair of nodes is smaller than the threshold, delete the edge shared by the pair of nodes.
4) Set the edge weight. Choose a positive function f (x) that is a monotonic increasing function of x. For the rest of the edges, set the weight as f (x), where x is the contact rate.

We use the algorithm for detecting communities in a 1000-taxi vehicular network based on the Shanghai taxi GPS traces. In **Fig. 3**, there are 33 communities in the whole network, and different colors represent different communities. The modularity of the detected communities is approximately 0.6, which indicates that the community structures are strong.

5 Verification

We verify our suggestion that the community structure in a vehicular network mainly accounts for f linear dependence of coded blocks received by a retrieving node. We first construct two synthetic network models that have very different properties in terms of community structure and then run simulations to study the impact of different degrees of community structure on linear dependence.

5.1 Synthetic Network Models

For comparison, we constructed another two different network models. In order to construct realistic network models, we referred to previous research on vehicular encounter models.

Encounters between vehicles could be modeled as a Poisson process. If we use $X_{i,j}(t)$ to denote the number of times vehicle v_i encounters vehicle v_j at time $t(X(0)=0)$, then $\{X_{i,j}(t), t>0\}$ is a Poisson process. Its rate $\lambda_{i,j}$ is equal to the reciprocal of average encounter time, which can be computed from real traces.

We could construct networks in which an arbitrary pair of nodes shares connections with intervals that follow an exponential distribution. In addition to the real vehicular network, we construct two synthetic networks with different degrees of community structure. The latter two networks have very different properties in terms of pair-wise connections, so we choose different rates to construct the two kinds of networks.

Uniform Network (UN): In this synthetic network, the inter-contact time of every pair of nodes follows a Poisson process whose rate is uniformly distributed over [0, 5]. In such a network, the community structure should not be clear.

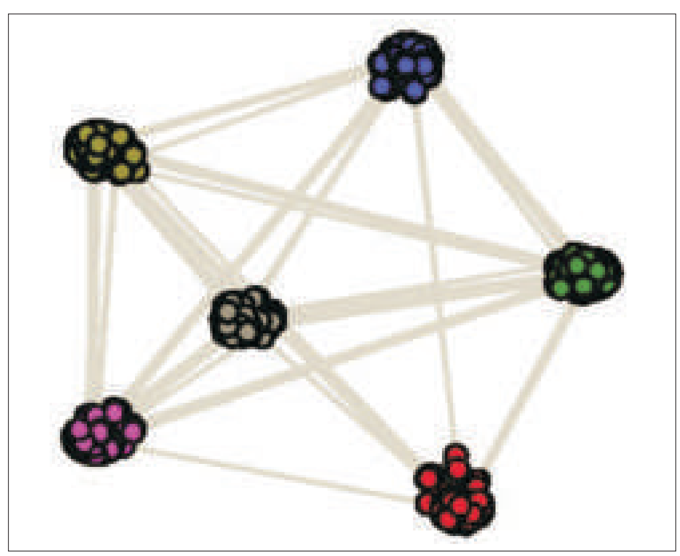
Synthetic Network (SN): This synthetic network should have a strong community structure. To construct a synthetic network, the nodes are divided into groups, each of which represents a community. A pair of nodes within the same group is generally assigned a shorter inter-contact time (a larger rate) than a pair of nodes from different groups. As a result, the nodes in the same group have more frequent contact. **Fig. 4** shows a synthetic network with six communities with 100, 150, 150, 200, 200, 200 nodes, respectively. The rate for a pair of nodes within the same group is uniformly distributed over [1, 5]. However, only 5% of node pairs from different groups can contact each other, and their rates are uniformly distributed over [1, 2]. As a result, the network has a clear community structure.

Real Network (RN): A real vehicular network is formed by using the Shanghai taxi GPS traces. Fig. 3 shows a real 1000-taxi vehicular network.

To determine the characteristics of these networks, we investigated the modularity of the three networks, and the results are shown in **Table 1**. The UN has the smallest modularity while both the SN and RN have much larger modularity. This comparison indicates that there are no clear communities in the UN, but there are clear community structures in the RN (Fig. 3) and SN (Fig. 4).

The SN is generated for comparison with the RN. Thus, the contact frequencies within the same community and across different communities are very different. From the modularity of

▲**Figure 3. A 1000-taxi vehicular network has 33 detected communities. Nodes in a community are the same color.**



▲**Figure 4. The community structures of a 1000-node network with clear community properties. This is an example of an SN.**

the SN, we can also determine that the SN has strong community structures (as expected).

5.2 Experiments

Our aim is to investigate the impact of community structure on the linear correlation of received codes. To compare data availability fairly, we fixed the communication cost during the code-distribution phase. The default setting for simulations is same as that in section 3.

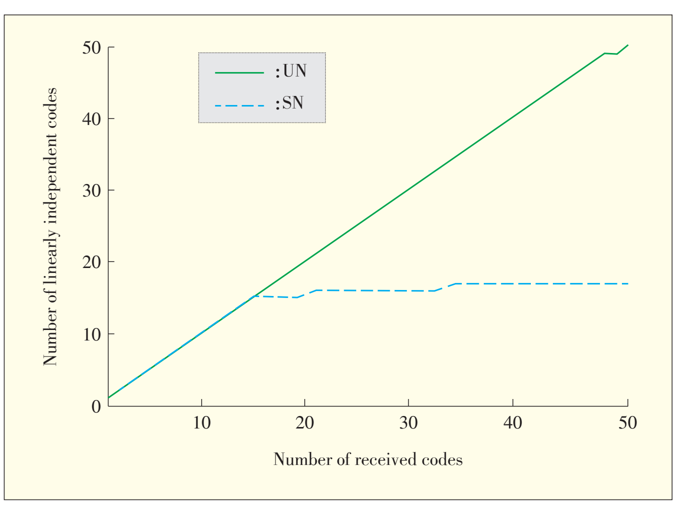
In **Fig. 5**, we show the change in the number of linearly independent codes as the number of received codes increases for the UN and SN. We found that the SN has a much smaller number of linearly independent codes than the UN. For example, when the number of received codes is 30, the UN almost has 30 linearly independent codes (i.e., no linear correlation between received codes) and the SN has only 17 linearly independent codes (i.e., 13 codes are redundant).

Before receiving 15 codes, the UN and SN almost share the same number of linearly independent codes, and both have no redundant codes. This can be explained by the detailed community structures. The average number of nodes in a community is $1000/6 \approx 167$, and the average number of source nodes within the community is $167 \times 50/1000 \approx 8$. Because data items in other communities are gradually propagated to the community that the retrieving node belongs to, it is reasonable that the number of different data items in a single community is larger than eight (here it is 17).

▼**Table 1. Modularity and communities for different kinds of networks**

	Modularity	Number of Communities
UN	0.008	5
RN	0.595	33
SN	0.756	6

RN: real network SN: synthetic network UN: uniform network



▲**Figure 5. Number of linear independent codes vs. the number of received coded blocks of UN and SN.**

We next explain why the community structure may lead to linear dependence of received codes. Within the short period of code retrieving, the retrieving node has probably received coded blocks, most of which are from nodes within its own community. As a result, there is probably a linear correlation between coded blocks if the number of coded blocks is much larger than the number of source nodes in its own community. (As previously mentioned, when nodes receive much more than 8, there is linear correlation.) Excessive coded blocks are not beneficial for the retrieving nodes to recover data items. The SN and the UN are two extremes, while the RN is similar to SN but has a weaker feature of community structure. This explains the linear dependence of received codes described in section 3. On the other hand, these redundant codes may be a waste of network resources.

6 Summary

In this paper, we have studied a network-coding-based approach to improving data availability in vehicular networks. Our empirical study with trace-driven simulations shows that performance can be significantly improved, but there is a serious issue: codes received by a retrieving node tend to be linearly correlated. Such linear correlation results in degraded data retrieval. To explain such an issue, we analyzed a large data set of taxi traces collected by thousands of taxis and reveal that a real vehicular network usually has clear communities. This community structure causes linear dependence between received codes. We verified this by devising SNs with dramatically different features of community structure and conducting experiments on the networks.

We also pointed out opportunities for improving the network-coding-based approach to improving data availability in a vehicular network by developing community-aware techniques for code distribution.

References

[1] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," in *Proc. 6th International Conference on ITS Telecommunications*, Chengdu, China, 2006, pp. 761-766. doi: 10.1109/ITST.2006.289012.

[2] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM MobiHoc' 04*, Tokyo, Japan, pp. 187-198. doi: 10.1145/989459.989483.

[3] U. Lee, E Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensing platforms," in *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 882- 901, Feb. 2009. doi: 10.1109/TVT.2008.928899.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H.

Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proc. ACM SenSys' 06*, Boulder, CO, USA, pp. 125–138. doi: 10.1145/1182807.1182821.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proc. ACM MobiSys' 08*, Breckenridge, CO, USA, pp. 29–39. doi: 10.1145/1378600.1378605.

[6] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. Ni, "Impact of traffic influxes: revealing exponential intercontact time in urban vanets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1258–1266, Aug. 2011. doi: 10.1109/TPDS.2010.176.

[7] T. Hara and S. K. Madria, "Data replication for improving data accessibility in ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 5, no. 11, pp. 1515–1532, Nov. 2006. doi: 10.1109/TMC.2006.165.

[8] P. Nuggehalli, V. Srinivasan, C. Chiasserini, and Ramesh R. Rao, "Efficient cache placement in multi-hop wireless networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 1045–1055, Oct. 2006. doi: 10.1109/TNET.2006.882863.

[9] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *Proc. ACM SIGCOMM' 02*, Pittsburgh, PA, USA, pp. 177–190. doi: 10.1145/964725.633043.

[10] S. M. Ross, *A First Course in Probability*, 8th ed. New Jersey, USA: Pearson Education, 2010.

[11] M. Fiore, C. Casetti, and C. Chiasserini, "Information density estimation for content retrieval in MANETs," *IEEE Trans. Mobile Computing*, vol. 8, no. 3, pp. 289–303, March 2009. doi: 10.1109/TMC.2008.110.

[12] D. Niu and B. Li, "Asymptotic optimality of randomized peer-to-peer broadcast with network coding," in *Proc. IEEE INFOCOM 2011*, Shanghai, China, pp. 1197–1205. doi: 10.1109/INFCOM.2011.5934898.

[13] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000. doi: 10.1109/18.850663.

[14] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sept. 2010. doi: 10.1109/TIT.2010.2054295.

[15] M. Wang and B. Li, "R2: random push with random network coding in live peer-to-peer streaming," *IEEE J. Selected Areas Commun.*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007. doi: 10.1109/JSAC.2007.071205.

[16] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006. doi: 10.1109/TIT.2006.881746.

[17] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, no. 10, Oct. 2008. doi: 10.1088/1742-5468/2008/10/P10008.

[18] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proc. 19th Int. Conf. World Wide Web*, Raleigh, NC, USA, 2010, pp. 631–640. doi: 10.1145/1772690.1772755.

Manuscript received: February 24, 2014

Biographies

Zhenni Feng (zhennifeng@sjtu.edu.cn) received her BEng degree in computer science and technology from Huazhong University of Science and Technology, Wuhan, China, in 2012. She is now a PhD student in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her research interests include participatory sensing, vehicular networks, and game theory.

Yanmin Zhu (yzhu@cs.sjtu.edu.cn) obtained his PhD in computer science from Hong Kong University of Science and Technology in 2007. He received his BEng. from Xian Jiao Tong University in 2002. He is an associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Prior to this, he was a research associate in the Department of Computing, Imperial College London. His research interests include ad hoc sensor networks, mobile computing, grid computing, and resource management in distributed systems. He is a member of the IEEE and IEEE Communication Society.

Qian Zhang (qianzh@cse.ust.hk) received her BS, MS, and PhD degrees in computer science from Wuhan University, China, in 1994, 1996, and 1999. She joined Hong Kong University of Science and Technology in September 2005 as an associate professor. She has published more than 200 papers in international peer-reviewed journals and conference proceedings on topics such as wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is a fellow of the IEEE.

Min Gao (mgao@cse.ust.hk) received his MS degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2005. He is a researcher at HKUST Fok Ying Tung Graduate School. His research interests include wireless sensor networks, RFID, and smart phones.

DOI: 10.3939/j.issn.1673-5188.2014.01.005
http://www.cnki.net/kcms/detail/34.1294.TN.20140320.1035.002.html, published online March 20, 2014

Networking-GPS: Cooperative Vehicle Localization Using Commodity GPS in Urban Area

Chisheng Zhang, Jiannong Cao, and Gang Yao
(Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China)

Abstract

A challenging issue in intelligent transportation systems (ITS) is to accurately locate moving vehicles in urban area. Considerable efforts have been made to improve the localization accuracy of standalone GPS receivers. However, through empirical study, we found that the latitude and longitude values generated by GPS receivers fluctuate significantly because of the multipath effect in urban areas. The relative distances between neighboring vehicles with similar GPS signal data in terms of satellite sets and signal strength are much more stable in such a scenario. In this paper, we propose a cooperative localization algorithm, Networking-GPS, to improve the accuracy of location information for vehicular networks in urban area using commodity GPS receivers. First, atom redundantly rigid graphs of vehicles are constructed according to the similarity of neighboring GPS data. Then, through rigidity expansion, local accuracy can enforce global accuracy. Extensive simulations based on the real road network and trace data of vehicle mobility demonstrate that Networking-GPS can improve the accuracy of the entire system.

Keywords

vehicular communication; cooperative localization; rigidity formation

1 Introduction

A vehicle localization system for urban areas is very challenging to design because of several rigorous requirements [1]. Localization accuracy should be satisfied by the users' applications, and the navigation system requires lane-level precision in order to differentiate between the different directions of lanes. GPS is the most feasible and robust solution for metropolitan-scale localization systems. However, GPS devices are inaccurate in urban areas because of severe multipath effect [2].

Our goal is to significantly improve the accuracy of localization information from GPS measurements by using commodity devices. We propose an accurate and cooperative localization algorithm, called Networking-GPS, for public transportation systems in urban areas. Networking-GPS is based on the observation that relative distances between neighboring vehicles with similar GPS signal data (in terms of satellite sets and signal strength) are much more stable. Networking-GPS constructs the atom redundantly rigid graphs with neighboring vehicles according to similarity of GPS data. Then, through rigidity expansion, local accuracy can enforce global accuracy. We have designed Networking-GPS using the commodity GPS devices without any modification of built-in GPS algorithms. The contributions of this paper are summarized as follows:

- We conduct experiments on several commodity GPS receivers to explore the relationship between the multipath effect and GPS measurements. We reveal that there are correlations between nearby GPS measurements under similar impact of multipath effect.
- To safeguard the accuracy of noisy GPS measurements against the multipath effect, we develop a rigid expansion algorithm by exploiting the signal similarity and ranging information from GPS measurements.
- We propose an accurate and cooperative localization algorithm, called Networking-GPS, for public transportation systems in urban areas. We run extensive simulations based on a real road network and trace data of vehicle mobility and determine that Networking-GPS improves the accuracy of the entire system.

The remainder of the paper is organized as follows. In section 2, we discuss related work. In section 3, we analyze the results from field testing and discuss opportunities to improve localization accuracy through neighboring GPS data. In section 4, we present the design of the Networking-GPS. In section 5, we evaluate the performance of Networking-GPS. Section 6 concludes the paper.

2 Related Work

2.1 GPS-Based Positioning Technology

A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include the time the message was transmitted, precise orbital information (the ephemeris and the general system health, and rough orbits of all GPS satellites (the almanac). The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite. These distances, along with the satellites' locations, are used with the possible aid of trilateration, depending on which algorithm is used, to compute the position of the receiver.

Three satellites might seem enough for positioning because space has three dimensions, and a position near the Earth's surface can be assumed. However, a very small clock error multiplied by the speed of light (at which satellite signals propagate) results in a large positional error. Therefore, receivers use four or more satellites to calculate the receiver's location and time.

2.2 Mitigation of Multipath Effect

In urban environments, multipath is the dominant factor contributing to inaccuracy [2]–[4]. Traditional approaches to multipath mitigation focus on detecting multipath-induced biases and alleviating their influence on the triangulation process. There are two categories of solutions to the multipath effect: Receiver autonomous integrity monitoring and peak separation. However, these two methods often fail to improve accuracy in urban areas because the model of multipath in the two solutions is a single reflection off of a single object.

An alternative approach to dealing with multipath in urban environments is a statistical model that accurately captures behavior of pseudo-range error distributions observed in real data, such as asymmetry, fat-tails, interdependence among errors, and dependence on multiple observable factors [5]. However, complexities associated with an accurate statistical model make estimation computationally challenging. Anti-multipath triangulation [6] is a proprietary technology developed to address this challenge. However, it is difficult to collect enough data to represent the complex multipath effect.

3 Motivation and Approach

In this section, we present and analyze the experimental results of GPS measurements using different commodity devices. Then, we discuss the opportunities to improve the localization accuracy through neighboring GPS data.

3.1 Preliminary Experiments

In a practical GPS localization system, GPS receivers calculate the positions after receiving enough valid GPS signals. When the results are ready, GPS chips write the values in the form of ASCII codes to interfaces such as serial or USB ports. The National Marine Electronics Association (NMEA) standard describes the interface between marine electronic equipment and is a common format for GPS result output. The information given by the GPS receiver includes latitude and longitude, altitude, speed, and time. In addition, NMEA allows proprietary sentences for private companies, which use these sentences as control information or output from GPS.

To examine the effect of multipath on GPS performance, we deployed four GPS receivers in different scenarios. Of the four GPS receivers, two have built-in SiRFstar-III GPS chips, and the other two have MTK GPS chips. After analyzing the GPS output, the SiRFstar-III-based GPS receivers follow the standard NMEA format. However, the MTK-based GPS receivers include some proprietary sentences. For ease of system design, we first reconfigured the MTK-based GPS receivers and converted the output format from the proprietary messages to standard NMEA sentences. Then, we fixed the GPS receivers on four corners of a square board. The distance between two GPS receivers can be ignored considering the size of the receiver. We set the sampling rate of GPS receivers to 1 s, which means that every 1 s, devices report their results to the interfaces.

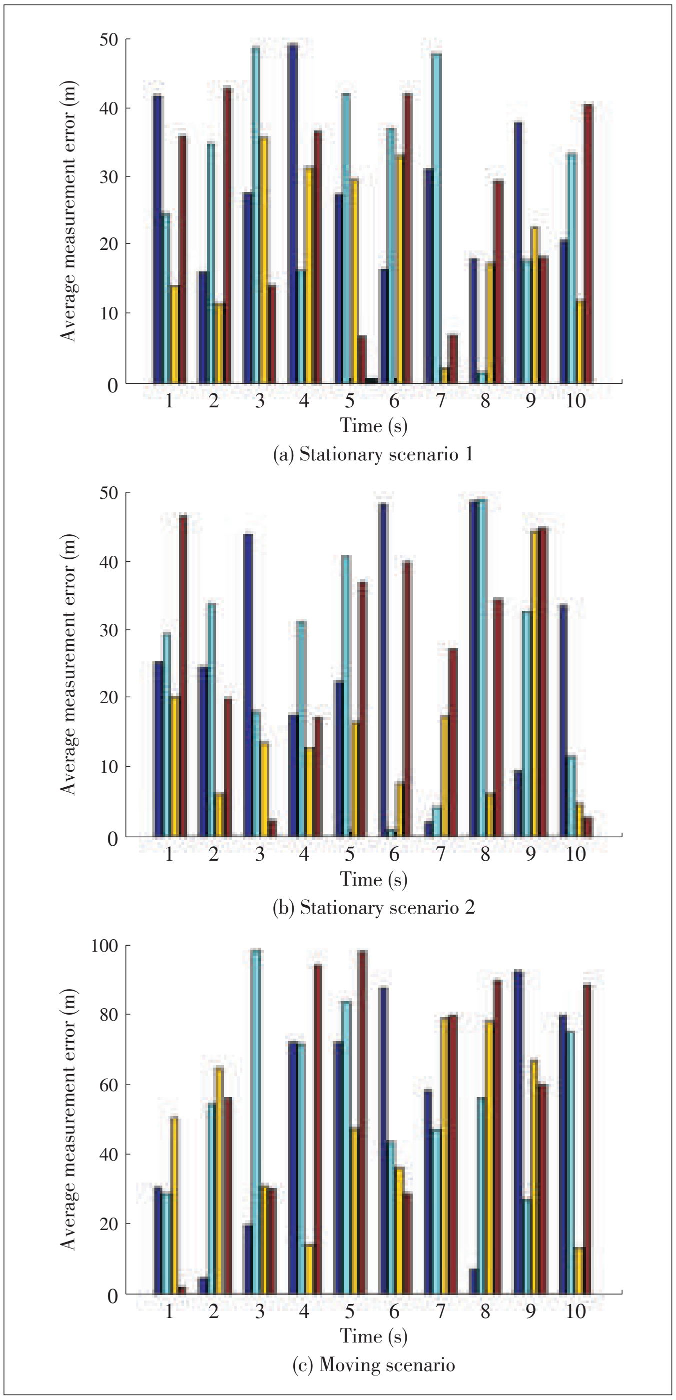
Both stationary and moving scenarios are considered in our experiments. In the stationary scenario, we placed the board in two different locations. First, we carried out the experiment in an open-sky environment, and then we placed the board outdoors near our office building. We collected all the GPS raw results from serial ports for further analysis. In the moving scenario, we put the board on the top of a taxi and obtained the GPS output during a round trip in the downtown district.

The results are shown in **Fig. 1**. The average GPS measurement errors are plotted against the experiment times. We found that GPS localization is relatively stable in an open-sky environment; however, the multipath effect severely degrades localization built-up environments and moving scenarios in urban areas. Localization errors can be up to hundreds of meters.

3.2 Similarity between GPS Signals

In the raw GPS result output, there is one type of sentence called GPGSV, which means GPS Satellite in View. This shows data about the satellites that the unit might be able to find based on its viewing mask and almanac data. It also shows current ability to track this data. One GPGSV sentence only can provide data for up to four satellites and thus there may need to be 3 sentences for the full information. The message format of GPGSV is $\$GPGSV,m,n,vv,i_1,e_1,a_1,s_1,\ldots,*CS$<CR><LF>, where T is the total number of messages, and M is message number (1 to 3). In every GPGSV message, there is up to four field group, as i_1,e_1,a_1,s_1, which stands for Pseudo Random Noise (PRN) code, Elevation in degrees, Azimuth in degrees and signal to noise ratio (SNR) from the respective satellites.

In the NMEA standard, the field called SNR is often referred to as signal strength. SNR is an indirect but more useful value than raw signal strength. SNR can range from 0 to 99 and is measured in decibels according to the NMEA standard. Zero is a special case and may be shown on satellites that are



▲**Figure 1. Localization variation with multipath effect.**

in view but not being tracked. Much effort needs to be made to reveal the correlation between SNR of GPS signals and the multipath effect. It has been reported that SNR depends on factors, such as GPS satellite transmit power, space loss and atmospheric attenuation, that are external to the receiver, and factors such as receiving antenna gain, tracking loop design and multipath effect, that are local. However, most of the work still tried to design the GPS algorithm to improve the performance of standalone GPS receivers.

Here, we utilize raw GPS data, especially SNR data and localization information, from nearby GPS receivers to improve the overall localization of multiple GPS receivers. Through the GPS raw data, we can obtain the corresponding SNR values of GPS signals from different satellites. To capture the spatio-temporal similarity in SNR variation from GPS signals, we devise a scalable representation of this information in form of a correlation matrix, $S(m,n)$. Each individual column corresponds to an SNR observation from GPS signal in a particular time, including the SNR values from m GPS satellites sensed by receivers. Each row is an n-element correlation vector, which means the signal variation from one particular GPS satellite in time domain. We captured the dominant variation patterns by using singular value decomposition (SVD) of the correlation matrix. SVD has the two main advantages: 1) it helps to convert high dimensional and high variable data set to lower dimensional space by exposing the internal structure of the original data more clearly; 2) it is robust to noisy data and outliers, which facilitates the further data processing.

By SVD, the correlation matrix, $S(m,n)$, can be represented as a product of three matrices: an orthogonal matrix U, a diagonal matrix A, and the transpose of an orthogonal matrix V. Therefore, we have $S = U A V^T$, where $U^T U = I = V V^T$, and A is a m-by-n matrix with r non-zero entries on its main diagonal containing the square roots of eigen values of matrix S in descending order of magnitude. The singular values of $A = \{a_1, a_2,...,a_r\}$ and the percentage of power captured by each eigen vector of matrix S is calculated by

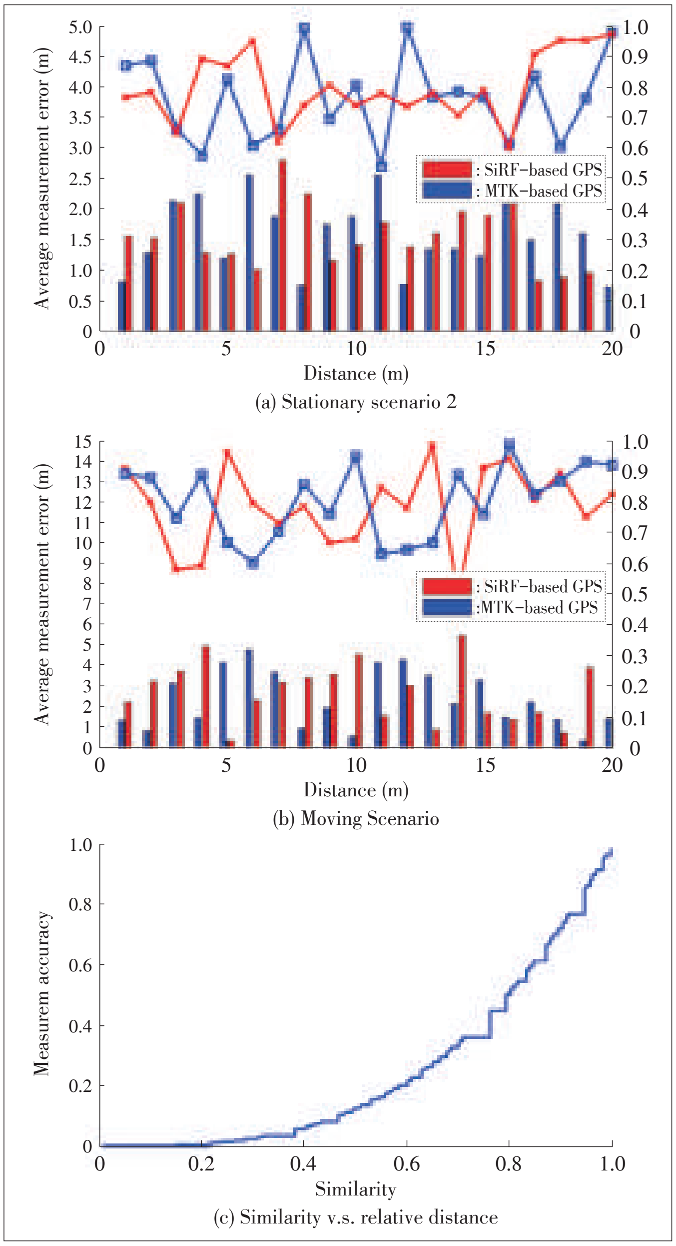
$$w_i = \sum_{i=1}^{k} a_i^2 / \sum_{i=1}^{rank(S)} a_i^2 \quad (1)$$

We use the eigen vectors of association matrix S to quantitatively measure the similarity between SNR values from different GPS receivers. For example, GPS raw data at two different locations, with respective eigen vectors as $X=\{x_1, x_2,...,x_{rx}\}$ and $Y=\{y_1, y_2,...,y_{ry}\}$. The signal similarity can be calculated by the weighted sum of pair wise inner product of their eigen vectors as

$$Sim(X,Y) = \sum_{i=1}^{rank(X)} \sum_{i=1}^{rank(Y)} w_{xi} \cdot w_{yi} \cdot \left| x_i \cdot y_i \right| \quad (2)$$

$Sim(X,Y)$ is quantitative measure index that shows the closeness of two GPS observation data in spatio-temporal dimension. The value of similarity lies between $0 \leqslant Sim(X,Y) \leqslant 1$. A higher value is derived from GPS data with similar correlation patterns. According to (2), we revisited the experiment results and checked the relationship between the similarity of SNR values and the relative distance (**Fig. 2**).

The relative distance has a strong correlation with the signal similarity. In Fig. 2a and Fig. 2b, relative distance between different static GPS receivers still varies because of severe multipath near tall buildings. However, when the signal simi-



▲**Figure 2. Correlation between distance variation with GPS signal similarity.**

larity increases, the relative distance between the considered GPS receivers decreases correspondingly. In Fig. 2c, the cumulative distribution function (CDF) between GPS signal similarity and measurement accuracy are shown through pair-wise comparison from our experiment results. We set the accuracy threshold for relative distance to 0.5 m, and we assume that the relative distance is zero when the calculated relative distance is not greater than 0.5 m. We find that in static scenario, when the similarity reaches to 0.9, the relative distance is almost equal to zero, with the possibility of 75%. From this observation, this method provides an accurate way to achieve local relative localization against the multipath effect based on GPS raw data.

4 Design of Networking-GPS

In this section, we present the cooperative vehicle localization algorithm, called Networking-GPS, and analyze the impact of the key parameters on the accuracy performance of the proposed localization algorithm.

4.1 Overview

The Networking-GPS algorithm can run on a central server or can be executed in each vehicle in a distributed way. We investigate the centralized version of Networking-GPS algorithm in this work, and we will study the distributed version in our future work.

In a vehicular network, vehicles with Internet connection have commodity GPS receivers for localization or navigation. GPS receivers calculate the positioning information and forward the raw GPS data to a central server. The raw GPS data is encapsulated in NMEA format. In the algorithm, we assume that the edge between two GPS receivers is a normal edge if a similarity index between them, calculated using (2), is greater than a threshold δ. Before describing the algorithm design in detail, we provide some preliminary knowledge about rigidity and redundant rigidity [7].

Definition 1: A rigid graph is an embedding of a graph in a Euclidean space that is structurally rigid. A graph is redundantly rigid if it is rigid after one (any one) of its edges is removed.

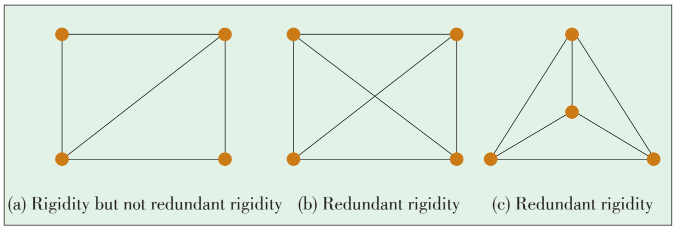
In our Networking-GPS algorithm, based on similarity correlation, we first construct the atomic redundantly rigid graphs and then expand the local graphs to a global graph by including some anchor points. After that, the local positioning information can be transformed into global localization information.

4.2 Construction of Atomic Redundantly Rigid Graphs

Because GPS signals have significant variation against time and space diversity, it is very difficult to form a global rigid graph for localization. To mitigate the impact of time and space diversity, we have to first construct some atomic redundantly rigid graphs based on the nearby GPS signals. We take four-vertex redundantly rigid graphs as the atomic graphs (**Fig. 3**). The procedure is shown in Algorithm 1.

In the beginning, the similarity between each pair of GPS receivers in the considered grid is calculated using (2). We selected the top four receivers according to the sum of similarity values between these receivers. If the similarity of the each pair among the selected four GPS receivers is greater than the threshold δ, the atomic redundantly rigid graph is found in the considered grid.

There are two important implications. First, due to the setting of similarity threshold δ, Algorithm 1 may find no satisfactory results for the considered grid. In the performance evalua-



▲**Figure 3. Rigidity and redundant rigidity.**

tion part, we will discuss the impact of threshold δ to the localization accuracy. The other implication is that there may exist more than one atomic redundantly rigid graph in one considered grid and Algorithm 1 only picks one from the existing redundantly rigid graphs.

Algorithm 1 Construction of atomic redundantly rigid graphs

Input:
A graph $G(V,E)$, the similarity threshold δ for defining normal links
Output:
Atomic redundantly rigid graph $G_r(V_r,E_r)$;
1: **for** $\forall v_i,v_j \in V$ **do**
2: $s_{i,j}=Sim(v_i,v_j)$,
3: **end for**
4: **for** $\forall \nu_i,\nu_j,\nu_k,\nu_l \in V$ **do**
5: $t_{i,j,k,l}=\sum v_{p,q} \in p,q\in\{i,j,k,l\}$,
6: **end for**
7: select $(a,b,c,d)=\text{argmax}\{t_{i,j,k,l}\}$, where $v_i,v_j,v_k,v_l \in V$
8: **if** $\forall s_{i,j}\geqslant\delta$, where $i,j\in(a,b,c,d)$ **then**
9: output G_r, where $V_r=a,b,c,d$ and $E_r=\{e_i,e_j | i,j\in\{a,b,c,d\}\}$
10: else
11: output $G_r=\emptyset$
12: **end if**

4.3 Expansion Algorithm of Redundantly Rigidity

After forming atomic redundantly rigid graphs, we can only obtain accurate relative positioning information between subsets of the GPS receivers. To map the relative information to global information, we need to expand the atomic redundantly rigid graphs to a larger area to cover some anchor nodes, such as fixed nodes or moving nodes with high localization accuracy.

Before presenting the detailed expansion algorithm, we introduce the following theorem [8], which shows the relationship between redundantly rigid graphs. From another perspective, it also provides an efficient method to do rigidity expansion.

Theorem 1: Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two redundantly rigid graphs with $|V_1 \cap V_2| >= 2$. Then,$|G_1 \cup G_2|$ is redundantly rigid.

Our expansion algorithm is a vertex-based solution that determines whether one vertex can be merged into a redundantly rigid graph. If the merged graph is still redundantly rigid, we select the redundantly rigid graph with the maximum cardinality and merge the vertex into the graph. If there is no merged graph that can merge the vertex to a larger redundantly rigid graph, then we define a weight function *RW(.)*, as shown in (3) to characterize the similarity between the vertex and other redundantly rigid graphs. According to the *RW* values, we can merge those vertexes to weighted redundantly rigid graphs.

$$RW(\nu,G,\delta)=\sum_{i=1}^{3}(Sim(\nu,\nu_i^G)-\delta)^3,$$
$$\text{where } \forall Sim(\nu,\nu_i^G)\Big|_{i=1,2,3} > Sim(\nu,\nu_j^G)\Big|\nu_j\in V^G-\{\nu_1,\nu_2,\nu_3\} \quad (3)$$

The entire algorithm is shown in Algorithm 2. In steps 1 to 4, vertexes are merged into larger existing redundantly rigid graphs. The merging process between redundantly rigid graphs is shown in steps 5 to 11. The weighted redundantly rigid graphs are constructed in steps 12 to 15.

Algorithm 2 Expansion algorithm of redundantly rigidity

Input:
A graph $G(V,E)$, a set of n redundant rigid components $G^r=\{G_1,G_2,\ldots,G_n\}$, the similarity threshold δ for defining normal linkds;
Output:
The maximum weighted redundantly rigid graph G^R;
1: **for** $\forall v\in G-\bigcup_{i=1}^{n}G_i$ **do**
2: select $G_m=\arg\max\{|G_i+v|\}$ and G_i+v is redundantly rigid
3: $G_m=G_m+v$
4: **end for**
5: $G_{rr}=\emptyset$
6: **for** $\forall G_i,G_j\in G_r$ **do**
7: **if** $|V_i\cap V_j|\geqslant 2$ **then**
8: $G_i\cup G_j\rightarrow G^{rr}$
9: $G^r=G^r-\{G_i-G_j\}$
10: **end if**
11: **end for**
12: **for** $\forall v\in G-\bigcup_{i=1}^{n}G_i^{rr}$ **do**
13: select $m=\arg\max\{RW(v,G_i^{rr},\delta)\}$
14: $G_m^{rr}=G_m^{rr}+v$
15: **end for**
16: output $G^R=\arg\max|G_i^{rr}|$

---

5 Performance Evaluation

We evaluate the performance of the Networking-GPS approach using real GPS samples. In our experiments, we first collected all the GPS raw data from the receivers and then established a simulation environment for the offline evaluation.

5.1 Raw Data Collection

We still used the two kinds of GPS receivers, as mentioned in section 3, with two different GPS chips. To facilitate data collection in vehicles, we built a small GPS collection box by packaging four functional elements: the GPS receiver, portable power bank, data processing unit and data storage unit. The data processing unit is responsible for receiving the GPS data and converting the raw data into a human-friendly format.

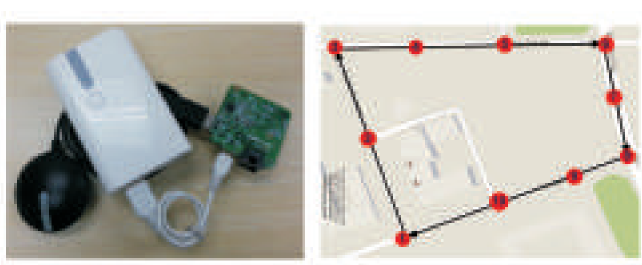
In our experiment, we adopted a small portable router TP-Link TL-WR703N as the data processing unit. TL-WR703N is a low cost 3G travel router and is very popular in developer community due to its open source from hardware to software. We flashed a compatible OpenWrt firmware to the router, which is an open source third-party router firmware system and compiled the necessary USB drivers and software package for the hardware. In our implementation, we utilized the TL-WR703N as the controller to receive and process the GPS raw data, and then forward it to the data storage unit. The key components of a GPS collection box are shown in **Fig. 4a**.

We mounted our small GPS collection boxes on the top of vehicles and recorded the raw GPS trace along the routes. At the end of the experiments, we exported all the data from the storage units. In addition, we obtained the ground truth of the localization information; we collected the latitude and longitude values of ten waypoints along the route, as shown in **Fig. 4b**, and the waypoints are numbered from 1 to 10.

5.2 Simulation Results

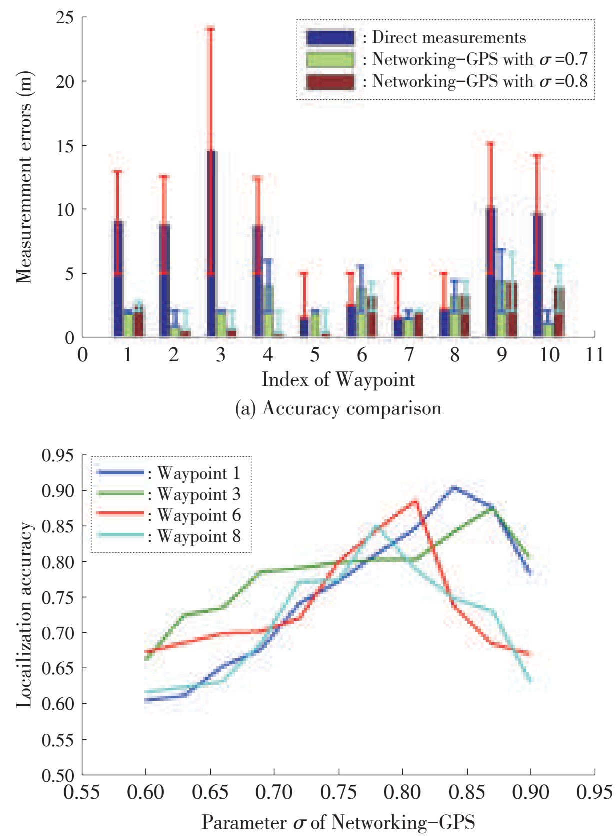
We ran offline simulations and processed GPS data traces from collection boxes. For ease of comparison, we only considered the GPS data collected around the ten waypoints in Fig. 4b. The accuracy comparison between direct measurements and our proposed Networking-GPS algorithm is shown in **Fig. 5a**. The blue bar represents the direct measurements from GPS receivers. The other two represent two different simulation results by choosing different threshold δ. The accuracy in terms of direct measurements is reduced because of the multipath effect, and the variations are also great. For example, at waypoint 3, the average localization error is nearly 15 m, and the variation is about 20 m. Networking-GPS provides the high accuracy and the localization information is much more stable than direct measurements.

In our Networking-GPS algorithm, we have a similarity threshold parameter δ that is involved in both Algorithm 1 and Algorithm 2. To indentify the impact of this parameter, we chose four waypoints to consider: 1, 3, 6, 8. Around those waypoints, we executed Networking-GPS algorithms under different thresholds δ, from 0.6 to 0.9 with step size of 0.3. The results are shown in **Fig. 5b**. We found that accuracy at different waypoints may present different trends. Therefore, the threshold δ should be adjusted along the entire route to achieve the best accuracy.



(a) Key components in a GPS collection box (b) Route and waypoints

▲**Figure 4. Scenario settings.**



(b) Impact of δ to localization accuracy

▲**Figure 5. Simulation evaluation.**

6 Conclusion

In this paper, we have developed Networking-GPS, a new cooperative vehicle localization algorithm, that uses commodity GPS in urban area. First, we showed that multipath effect severely degrades localization performance of GPS receivers. Then, we identified the correlation between the similarity of

SNR values from different GPS satellites and relative distance between different GPS receivers. Based on the observation, we designed the Networking-GPS algorithm. Our evaluation based on real GPS traces shows that Networking-GPS is highly accurate despite the multipath effect. In our future work, we will investigate the distributed version of Networking-GPS and optimal parameter adjustment algorithm.

References

[1] H. Zhu, Y. Zhu, M. Li, and L. Ni, "Hero: Online real-time vehicle tracking in shanghai," in *INFOCOM 2008, The 27th Conference on Computer Communications*, Phoenix, AZ, 2008, pp. 942-950. doi: 10.1109/INFOCOM.2008.147.

[2] N. Agarwal, J. Basch, P. Beckmann, P. Bharti, S. Bloebaum, S. Casadei, A. Chou, P. Enge, W. Fong, N. Hathi et al., "Algorithms for gps operation indoors and downtown," *GPS solutions*, vol. 6, no. 3, pp. 149-160, 2002. doi: 10.1007/s10291-002-0028-0.

[3] A. Bilich and K. M. Larson, "Mapping the gps multipath environment using the signal-to-noise ratio (snr)," *Radio Science*, vol. 42, no. 6, 2007. doi: 10.1029/2007RS003652.

[4] A. Bilich, K. M. Larson, and P. Axelrad, "Observations of signal-to-noise ratios (snr) at geodetic gps site casa: Implications for phase multipath," in *Proceedings of the Centre for European Geodynamics and Seismology*, vol. 23, pp. 77-83, 2004.

[5] W. Sun and M. G. Amin, "Maximum signal-to-noise ratio gps anti-jam receiver with subspace tracking," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP' 05). IEEE International Conference on* (*vol. 4*), Philadelphia, PA, 2005, pp. iv-1085. doi: 10.1109/ICASSP.2005.1416201.

[6] J. Stone and M. Chansarkar, "Anti-multipath triangulation (amt) for positioning in dense urban environments," in *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*, Long Beach, CA, 2004, pp. 1165-1168.

[7] A. Franchi and P. R. Giordano, "Decentralized control of parallel rigid formations with direction constraints and bearing measurements." in *CDC*, T Okada, HL Tang, 2012, pp. 5310-5317.

[8] Z. Yang, L. Jian, C. Wu, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, p. 26, 2013. doi: 10.1145/2422966.2422983.

Manuscript received: February 21, 2014

Biographies

Chisheng Zhang (cscszhang@comp.polyu.edu.hk) is currently a PhD candidate of the Department of Computing in the Hong Kong Polytechnic University. He received his B.Sc. and M.Sc. degree in 2005 and 2008 respectively in pattern recognition and intelligent system from Northwestern Polytechnical University in Xi'an, Shaanxi. His research interests include multi-hop wireless networks, vehicular communication systems and cooperative communications.

Jiannong Cao (csjcao@comp.polyu.edu.hk) is currently a chair professor and head of the Department of Computing at Hong Kong Polytechnic University. He received his BSc degree in computer science from Nanjing University, China, in 1982. He received his MSc and PhD degrees in computer science from Washington State University in1986 and 1990. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He has co- authored four books, co-edited nine books, and published more than 300 technical papers in major international journals and conference proceedings. He has directed and participated in numerous research and development projects and, as a principal investigator, obtained more than HK$25million in grants. He is the chair of Technical Committee on Distributed Computing, IEEE Computer Society; a senior member of IEEE; a member of ACM; and a senior member of China Computer Federation. He has been an associate editor and member of editorial boards of many international journals. He has been the chair and a member of organizing/program committees for many international conferences.

Gang Yao (csgyao@comp.polyu.edu.hk) is currently is currently a PhD candidate in the Department of Computing, Hong Kong Polytechnic University. He received his BSc degree in computer science from Wuhan University, China. His research interests include seamless communication and mobility management, mobile computing, wireless networks and security.

ZTE Achieves Second Place in World Intellectual Property Organization Patent Table

March 18, 2014—ZTE Corporation achieved second place in the World Intellectual Property Organization's annual table of patent applicants, strengthening the company's position as one of the world's leading technology innovators.

With 2309 filings under the Patent Cooperation Treaty (PCT), ZTE was second only to Panasonic. In 2011 and 2012, ZTE ranked number one.

"ZTE's growing portfolio of intellectual property assets is providing strong support to the company's development of new technologies," said Guo Xiaoming, chief legal officer at ZTE. "PCT patents are the most valuable IP assets of a company. ZTE is committing tremendous resources to patent applications, as the company seeks to play a leading role in the global technology industry by shaping the development of essential underlying technologies."

According to WIPO, the number of patent applications exceeded 200,000 globally for the first time in 2013. China was one of the top three countries making patent applications.

ZTE has now built up strong positions in the development of key technologies such as operating systems, databases, mobile devices, applications, security solutions and semiconductors.

ZTE has filed applications for more than 50,000 patents globally, and more than 16,000 of these have been granted. ZTE is a global leader with more than 800 essential patents on 4G LTE standardization. ZTE will continue to strengthen its portfolio of patents to attain the leading position in smart terminals, optical networking, cloud computing, big data and 4G LTE and will invest in next-generation technologies such as 5G. (ZTE Corporation)

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 006
http://www.cnki.net/kcms/detail/34.1294.TN.20140326.1652.001.html, published online March 26, 2014

Anatomy of Connected Cars

Mario Gerla[1], **Giovanni Pau**[1,2], **and Rita Tse**[1,3]
(1. UCLA-MPI Joint Research Laboratory in Ubiquitous Computing, University of California, Los Angeles, California, USA;
2. Universite' Pierre and Marie Curie-LIP6, Paris, France;
3. Computing/Computer Studies Program, Macao Polytechnic Institute, Macao, China)

Abstract

The National Highway Traffic Safety Administration in the US and the European Commission are drafting a regulatory framework that will make the goal of connected vehicles possible by 2020. Control, embedded systems, and communication technologies have developed over the past 10 plus years and are approaching maturity. These will spark a revolution in how we approach driving. Cars will no longer need human drivers; they will be connected and exchange information about navigation, road hazards, traffic conditions, and safety. Travelers will be connected more than ever. Today's car will become tomorrow's office and the act of driving will become a leisure activity rather than a necessity. The emerging Internet of Vehicle enables application scenarios unimaginable just few years ago. The main challenges are Internet access spectrum scarcity, mobility, intermittent connectivity and scalability. In this article, we discuss the evolution from intelligent vehicle grid to autonomous, internet-connected vehicles and vehicular cloud.

Keywords

connected vehicles; Internet of Vehicles; name data networking

1 Introduction

Traditionally, a vehicle has been considered an extension of human ambulation and is easily controlled by the driver's commands. Recent advances in communications, controls and embedded systems have changed this model, paving the way for the intelligent vehicle grid. The car is now a formidable sensor platform, absorbing information from the environment (and from other cars) and feeding it to drivers and infrastructure in order to assist with safe navigation, pollution control, and traffic management. The next step in this evolution is just around the corner: the Internet of Connected Autonomous Vehicles. Pioneered by Google car, the Internet of Vehicles (IoV) will be a distributed transport fabric capable of making its own decisions about driving customers to their destinations. Like other important instantiations of the Internet of Things (IoT) (e.g., smart building), IoV will have communications, storage, intelligence, and learning capabilities in order to anticipate customer intentions. The concept that will help in the transition to IoV is vehicular cloud, the equivalent of Internet cloud for vehicles. Vehicular cloud will provide all kinds of the services required by connected vehicles.

ATOS and RENAULT through the UPMC Chair on Smart Mobility, the MPI Research Fund, and the National Science Foundation through the GREEN-CITY project support this work.

The urban fleet of vehicles is rapidly evolving. Currently, it is a collection of sensor platforms that provide information to drivers and upload filtered sensor data (e.g., GPS location, and road conditions) to the cloud (**Fig. 1**). However, it will evolve to become a network of connected vehicles that exchange their sensor inputs with each other in order to optimally perform a well-defined function. This function, in the case of autonomous cars, is prompt delivery of the passengers to destination with maximum safety and comfort and minimum impact on the environment. In other words, one is witnessing in the vehicle fleet the same evolution from Sensor Web (i.e., sensors are accessible from the Internet to get their data) to IoTs. (The components with embedded sensors are networked and make intelligent use of these sensors.)

In an intelligent home, the IoT is formed by a myriad of sensors and actuators that cover the house internally and externally. The IoT can manage utilities in the most economical way, with maximum comfort to residents and almost no human intervention. Similarly, in the modern energy grid, the IoT formed by large and small components can manage power loads in a safe, efficient manner with operators playing the role of observers.

In the vehicular network, like in all the other IoTs, when human control is removed, autonomous vehicles must efficiently cooperate so that traffic flow on roads and highways is smooth. It is predicted that vehicles will behave much better than drivers, and more traffic will be handled with fewer delays, less

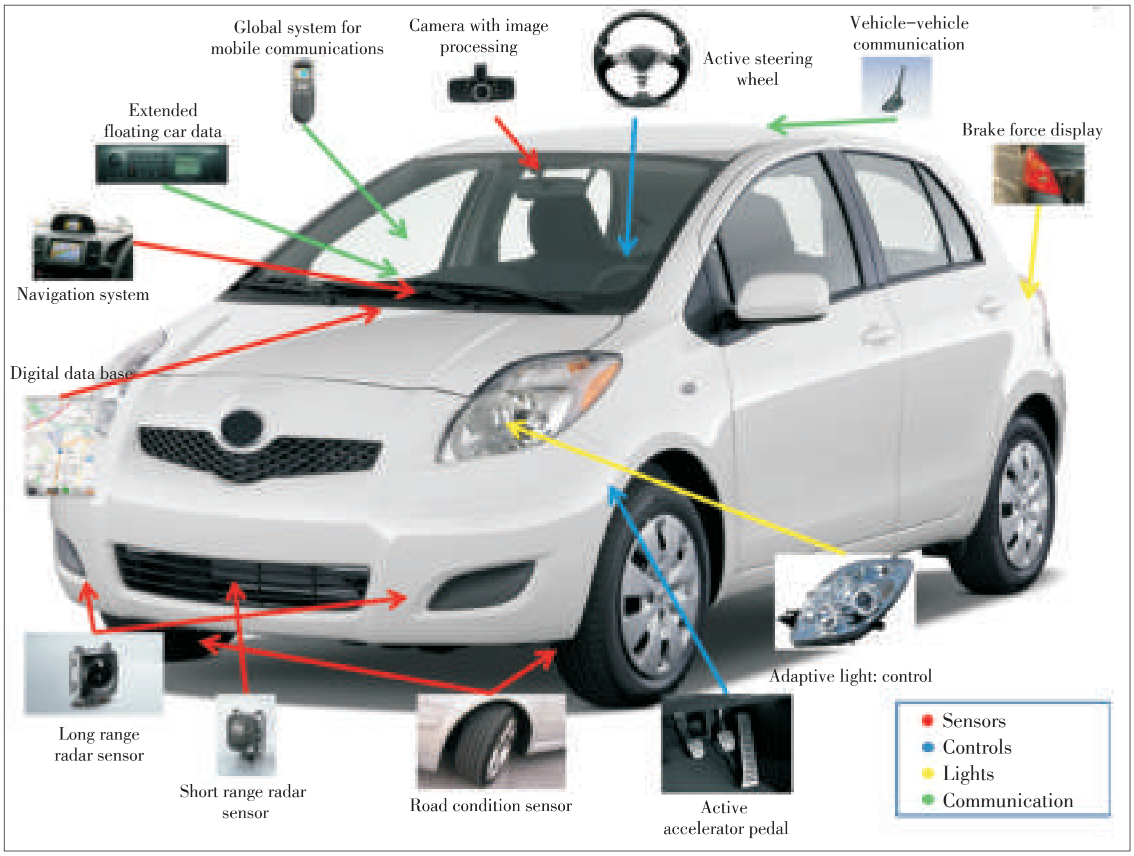


Figure 1.
Vehicles are mobile sensors that produce massive amounts of data.

pollution, and more driver and passenger comfort. However, the complexity of distributed control of hundreds of thousands of cars cannot be taken lightly. In the event of a natural disaster, vehicles must be able to coordinate to rapidly evacuate critical areas in an orderly manner. This requires efficient communication between vehicles and the ability to discover the location of needed resources (e.g., ambulances and police vehicles, information about escape routes, and images about damage that must be avoided).

An efficient communication and distributed processing environment can be provided by a new network and compute paradigm specifically designed for vehicles: vehicular cloud. This mobile cloud provides several essential services—from routing, content search, spectrum sharing, dissemination and attack protection to connected vehicle applications via standard, open interfaces that are shared by all auto manufacturers.

This article discusses the evolution from intelligent vehicle grid to autonomous, Internet-connected vehicles and vehicular cloud. In particular, we highlight the advantages of the Internet of Autonomous Vehicles and discuss related challenges.

2 Characteristics of IoV Applications

Applications in vehicle communications have ranged from safety and comfort to entertainment and commercial services. This section discusses four noticeable characteristics in emerging vehicle applications and offers a vision for intelligent vehicle grid and its impact on the autonomous vehicle. Specifically, we base our observations on actual experiments run in Macao, where we operate an urban sensing testbed, and at University of California, Los Angeles (UCLA), where we operate a vehicular V2X testbed [1], [2]. We observed the following information characteristics that are common across a large number of application scenarios.

2.1 Information Characteristics of IoV Applications

Vehicles are data "prosumers"; that is, they have a plethora of sensors that produce a great amount of content, and at the same time, these vehicles consume content from other cars and the Internet. There are several common properties in terms of locality and lifetime. Here, we identify three general properties that are common to many application scenarios:

1) Limited lifetime. Vehicle-oriented content has its own temporal scope of validity. This implies that the content must be available during its lifetime. For example, road congestion information may be valid for only a few minutes or an accident warning must remain as long as road is not cleared.
2) Spatial validity. Car-generated content has a spatial scope of utility to information consumers. In safety applications, a speed warning near a blind intersection is only valid to vehicles approaching to the intersection, say within few hundred meters. In a parking scenario, the information has inherent

local validity because users are generally interested in parking close in rather than far away. A similar concept applies to a plethora of other vehicular applications, including advanced navigation systems.

3) Interest locality. This indicates that nearby vehicles represent the bulk of potential content consumers. This concept is further extended so as to distinguish the scope of consumers. For instance, all the vehicles in the vicinity want to receive safety messages, while only a fraction of vehicles are interested in commercial advertisements.

Time-space validity of the data implies the scalability of the data collection/storage/processing applications, since old data is discarded. It also implies that the data should be kept on the vehicles rather uploaded to the internet, leading to enormous spectrum savings. This property will be key to the scalability for the Internet of vehicles concept, given the huge amount of data collected by autonomous vehicle sensors [3].

2.2 Information-Centric Networking

Vehicle applications are mainly concerned with content itself, not its provenance. This memory-less property is characteristic of VANETs. To check traffic congestion on the fixed internet, one visits a favorite service site. The site's URL guarantees access to ample, reliable information. In contrast, vehicle applications flood query messages to a local area, not to a specific vehicle, and accept responses regardless of the identity of the content providers. In fact, the response may come from a vehicle in the vicinity that has in turn received such traffic information indirectly from neighboring vehicles. In this case, the vehicle does not care who started the broadcast. This characteristic is mainly due to the fact that the sources of information (vehicles) are mobile and geographically scattered. A typical example is an application that is interested in determining if there are any available roadside parking spaces in a specific area. With IP, a server infrastructure is required to store and record the available parking spots [4]; however, with the ICN paradigm, it is possible to directly query the network for available spots [5], [6].

We expect information-centric networking to play a major role in the management and control of autonomous vehicles. There are two reasons for this: first, the autonomous vehicle will travel at high speeds and short distances from neighbors (on highways) and must have up-to-date information from surrounding vehicles of up to several kilometers away in order to maintain a stable course [3]. Thus, in the content-centric networking style, the vehicle periodically sends interest messages to receive position, speed, and direction information from the rest of the fleet. Second, in the case of an accident ahead, the vehicle must alert the driver (who may have their attention elsewhere) so that they have the option of manually intervening. To prepare the driver to take over, the vehicle retrieves photos and possibly video of the accident scene from the cameras of the vehicles facing the accident. Content-centric networking allows access to the best cameras with the needed data, without prior knowledge of the cars that are offering the data.

2.3 Vehicular Sharing of Sensory Data

Emerging vehicle applications consume a huge amount of sensor data in a collaborative manner. That is, multiple sensors installed on vehicles record a myriad of physical phenomena. Vehicle applications collect sensor records, even from neighboring vehicles, to produce value-added services. The CarSpeak application [3], for example, enables a vehicle to access sensors on neighboring vehicles in the same way that the vehicle can access its own sensors. The vehicle then runs an autonomous driving application using the sensor collection without knowing who produced what. Extending the sensorial capabilities of connected cars enables situational awareness in an "ahead of time" fashion, thus allowing drivers and autonomous vehicles to identify road hazards. Proactive identification of potential road dangers is essential to preventing accidents. For example, if we know that just around a sharp corner pedestrians are j-walking, then a potentially deadly accident could be averted. In the general perspective of an intelligent transport system, vehicles exchange traffic congestion and road condition messages to construct an up-to-date traffic and road condition database from which best path to (local) destinations are computed. Collaboration in the sharing and processing of sensor data will be one of the strong advantages of autonomous vehicles. Continuous sharing of position data is essential to guarantee the stability of the autonomous fleet. The crowdsourcing of road conditions, such as poor pavements, obstacles, and accidents, using the collection of available sensors will allow smooth driving, even in perilous conditions. Moreover, the collective tracking of available channels using sophisticated on board radios will allow careful mapping of the available spectrum, enabling the efficient communications required for fleet situation awareness and content downloading to "passive" drivers.

2.4 Intelligent Vehicle Grid and Vehicular Cloud

Vehicles have sensors that generate copious amounts of data every second. At the same time, the road has smart components, RFID tags, and embedded microcontrollers [7], [8]. These "things" constitute a vehicle grid, i.e., an intelligent road infrastructure analogous to the energy grid for intelligent power generation and distribution. The vehicular grid will have the resources to support services and applications for connected vehicles and autonomous vehicles alike. The various things in the vehicular grid will evolve into the vehicular cloud that provides the computing and communication environment for the Internet of Vehicles. Vehicles become service providers and consumers as well as the infrastructure without an explicit addressing of the resources (i.e. retrieving information from deep-inside the Internet or from a car will be transparent for the user).

The main beneficiaries of vehicular cloud architecture will be autonomous vehicles that drive themselves without human intervention. Such vehicles must be capable of sensing their surroundings and locating routes as well as obstacles. An advanced autonomous car processes all sensory data gathered onboard and from other vehicles [3], identifies appropriate paths, and constructs a decision tree in order to avoid obstacles on these paths. The vehicular cloud will provide the ideal system environment for the coordinated deployment of the sensor aggregation, fusion and database sharing applications required by the IoVs.

3 IoV Connectivity Challenges

There are several challenges to be overcome before the IoV moves from the research prototype stage to commercial deployment. In particular, IoV nodes need to cope with Internet access, spectrum scarcity, mobility, intermittent connectivity and scalability. In the remainder of this section, we briefly report on recent advances in internet access as well as mobility management achieved through the use of multiple wireless technologies and substituting IP with the named data networking (NDN) information paradigm [9].

3.1 In-Vehicle Connectivity

Several wireless technologies will be required in order to support IoV at scale. The nature of the information (i.e. spatial, temporal, and interest locality), the requested bandwidth, and channel contention mean that a single solution cannot be relied on. Car manufacturers, network operators, and cellular providers will need to provide a plethora of options for the connected vehicle [10].

In-vehicle connectivity will be guaranteed by wireless technologies, including IEEE802.11p, Wi-Fi, and 3G/LTE. The networking stack on the vehicle head unit will be in charge of selecting the appropriate communication channel(s) according to current status, user preferences, and application needs. At scale, metro-scale Wi-Fi will be crucial for reducing the load on the cellular infrastructure, especially during rush hours when vehicular density is very high [11]–[13]. In the initial phase, we expect IEEE802.11p (DSRC) to play a major role in V2V applications, such as safety and local information retrieval. Massive deployment of DSRC roadside infrastructure is not expected any time soon because of the massive investment required. In contrast, there are many existing city-scale Wi-Fi deployments. In initial deployment scenarios, high-bandwidth Internet connectivity will be facilitated by cellular and Wi-Fi Infrastructures. Mobile operators are also running out of cellular spectral resources and are encouraging their customers to use the extensive Wi-Fi network wherever possible. This trend is particularly pronounced in France, where all mobile operators offer free Wi-Fi traffic and other incentives to customers who accept to be part of a community of Wi-Fi Hot-Spots[1]. The Wi-Fi protocol, however, was not designed to handle high-speed mobility and frequent connection/disconnections of moving vehicles. The initial connection (attach) time required to access a Wi-Fi hot-spot is prohibitive for a moving car [14]. Even at a moderate urban speed of 35 km/h, the time required to connect, authenticate, acquire an IP, and finally transmit and receive packets on a standard Wi-Fi network is generally too long to be useful [14]. The car will likely leave the coverage area before it has any chance to communicate. An enabler to the IoV vision is the Wi-Fi fast-attach.

An initial approach to the quick access point attach was explored in Quick Wi-Fi [14]; however, the proposed solution is limited. It relies on the wireless interface monitor mode that limits the operation to the basic rate (usually 1 Mbit/s). It lacks portability because it uses hardware-dependent primitives, and it does not support any authentication method. Recent research on the fast-attach mechanism done at UCLA and at LIP 6 in Paris has shown promise in terms of resolving the Wi-Fi initial-attach problem. We redesigned the software component that manages Wi-Fi connectivity in Linux/Androd systems, i.e., the WPA supplicant, in order to achieve quick initial connections and integrate the most common authentication methods. WPA fast is a redesigned WPA supplicant component that uses advanced scanning algorithms that can learn from the environment in order to estimate the amount of time spent channel scanning. Furthermore, it combines the attach process with the authentication process, and this reduces connection time. Although detailed description of the WPA-fast protocol is out of the scope of this paper, preliminary experiments performed at UCLA show that WPA-fast cuts the initial attach time needed for a hot-spot connection by one order of magnitude compared to today's standard WPA. In particular, we drove about two miles across Los Angeles departing from the UCLA campus across a residential area. We tried to connect to any Open or Eduroam authenticated network (Eduroam authentication is based on IEEE802.1x + Radius). WPA Fast was constantly able to connect showing a one order of magnitude improvement. In particular, the 80th percentile connection latency for a vehicular Wi-Fi client reduced from 10 s to 0.6 s (**Fig. 2**).

Although WPA-fast is still far from being finalized, it produces encouraging results. Once it has been finalized and extensively tested in both Android and Linux devices, it will contribute to removing one of the main roadblocks to scalable, inexpensive high-speed connectivity in connected vehicles.

3.2 Named Data Networking

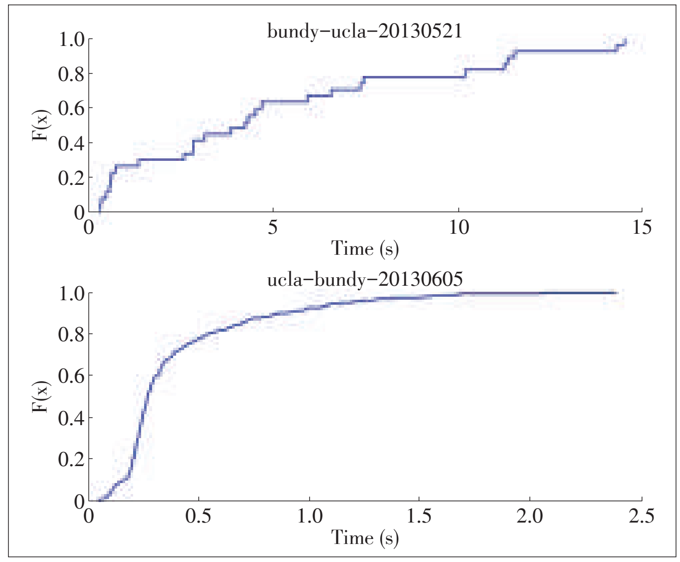
Scalability issues have traditionally affected mobility over IP. When the IP address changes frequently, such as in community Wi-Fi, mobile IP protocols suffer from high overhead in

[1] France network operators encourage residential users to offer a slice of their home-connection to other fellow customers in exchange of the similar privileges. The authentication is operator controlled. This "community" based approach allowed France network operators to grow their WiFi networks quickly and substantially at no cost.

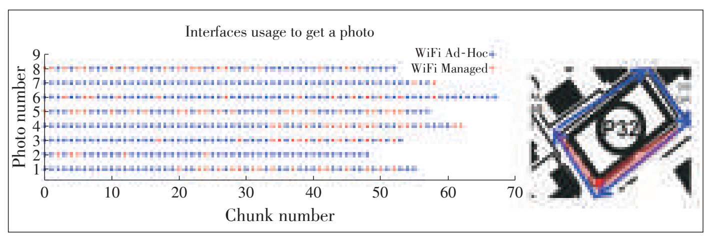
both the registration process and routing process. Recently, Van Jacobson et al. [9] proposed a radical change to the Internet architecture and shifted the paradigm from a node-centric network to a content-centric network. They observed a huge gain in efficiency by using distributed in-network storage. The NDN architecture proposes to directly address content at the network layer using hierarchical names. The IP layer is removed and substituted by the object name. NDN revolves around the pull model in which consumers are first-class citizens that request a content piece though a small special packet called interest-packet. For each interest issued by a consumer, the network searches for a matching content unit. If found, the content is delivered to the consumer as an interest response. Routing and forwarding are re-engineered to work on the NDN namespace [9]. An initial version of the NDN forwarder was released in 2009, and at present, several state-of-the-art applications have been developed for the NDN architecture, including multiuser video conferencing, web browser, multi-room chat programs, and IoT applications [15], [16]. In a connected car scenario, NDN naturally addresses mobility. When a car moves from an access point to another, it only takes the consumer the cost of resending the last not satisfied interest to restart the communication from where it was left. Network caching prevents wasteful retransmission, and exploiting the broadcast in the wireless segment enables the potential of the shared medium to be fully exploited.

We designed and developed NDN for vehicular scenarios [6]. The protocol was evaluated using a small testbed and simulations. The actual experiments were carried at UCLA using ten cars. We identified several application scenarios, including real-time picture transmission and retrieval of real-time traffic information from the road [6]. A detailed report on the protocol design and experimental setup can be found in [6] and [17]. In a mobility scenario, it was possible to send medium-sized pictures (~100KB) using two different wireless connections, a Wi-Fi hot spot, and an ad-hoc connection through a moving relay car. In **Fig. 3**, the color of the dots offers a qualitative hint on how the data was transferred across different interfaces. In contrast with TCP/IP V-NDN does not need any special mechanism to address the client mobility, as it is part of the architecture design to the benefit of simplicity and scalability.

The experimental results collected using the UCLA testbed provided a number of insights on the potential advantages provided by Named Data in vehicular environment. In particular, the in-network caching features of V-NDN and the per-packet enable to take full advantage of mobility and to cope with intermittent connectivity and disruption. The in-network cache was able to satisfy about one third of the total requests thus reducing the network load and the application latency. Furthermore, in the experiment above was possible to use either one or the other interface or in some cases both of them seamlessly. NDN interest-data mechanism allows to change network interface at with a per-interest granularity. This approach, in contrast with Mobile IP, requires no triangular routing and no additional network components.



▲ **Figure 2. WPA Fast performance in an open network. The chart shows the cumulative distribution function of the time needed to connect to a new access point.**



▲**Figure 3. Real-time retrieval via V-NDN.**

4 Final Remarks

In this paper, we introduced the Internet of Vehicles, which is poised to become a reality with the heavy deployment of connected vehicles (as mandated by the US Department of Transportation) [18]. We identified the main characteristic of the IoV information flow and exposed two of the roadblocks to its deployment, arguing that the ability to exploit the community Wi-Fi connectivity and leave shift from a node-centric paradigm to a content centric paradigm can accelerate deployment and reduce total costs.

References

[1] G. Pau and R. Tse, "Challenges and opportunities in immersive vehicular sensing: lessons from urban deployments," *Image Communication*, vol. 27, no. 8, pp. 900–908, Sept. 2012. doi: 10.1016/j.image.2012.01.015.

[2] M. Gerla, , J.-T. Weng, E. Giordano, and G. Pau. "Vehicular testbeds-model validation before large scale deployment," *Journal of Communications*, vol. 7, no. 6, pp. 451–457, Jun. 2012. doi: 10.4304/jcm.7.6.451–457.

[3] S. Kumar, L. Shi, S. Gil, N. Ahmed, D. Katabi, and D. Rus, "CarS-peak: a content-centric network for autonomous driving," in *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 259-270, Oct. 2012. doi: 10.1145/2377677.2377724.
[4] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proc. 8th Int. Conf. Mobile Systems, Applications, and Services*, San Francisco, CA, USA, Jun. 2010, pp. 123-136. doi: 10.1145/1814433.1814448.
[5] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "VANET via Named Data Networking," in *Proc. 2nd IEEE Workshop on Name Oriented Mobility (NOM), INFOCOM'14*, Toronto, Canada.
[6] D. Pesavento , G. Grassi, C. E. Palazzi, and G. Pau, "A naming scheme to represent geographic areas in NDN," *Wireless Days (WD), 2013 IFIP*, Valencia, Spain, Nov. 2013, pp. 1-3. doi: 10.1109/WD.2013.6686541.
[7] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for 'Smart Dust'," in *Proc. 5th Annu. ACM/IEEE Inter. Conf. Mobile Computing and Networking*, Seattle, WA, USA, Aug. 1999, pp. 271-278. doi: 10.1145/313451.313558.
[8] E.-K. Lee, S. Yang, S. Y. Oh, and M. Gerla, "RF-GPS: RFID assisted localization in VANETs," in *IEEE 6th Int. Conf. Mobile Adhoc and Sensor Systems*, Macau, China, Oct. 2009, pp. 621-626. doi: 10.1109/MOBHOC.2009.5336946.
[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerging Networking Experiments and Technologies*, Rome, Italy, Dec. 2009, pp. 1-12. doi: 10.1145/1658939.1658941.
[10] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *Pro. 8th Int. Conf. Mobile Systems, Applications, and Services*, San Francisco, CA, USA, Jun. 2010, pp. 209-222. doi: 10.1145/1814433.1814456.
[11] P. Deshpande, X. Hou, and S. R. Das, "Performance comparison of 3G and metro-scale WiFi for vehicular network access," in *Proc. 10th ACM SIGCOMM Conf. Internet Measurement*, Melbourne, Australia, Nov. 2010, pp. 301-307. doi: 10.1145/1879141.1879180.
[12] X. Hou, P. Deshpande, and S. R. Das, "Moving bits from 3G to metro-scale WiFi for vehicular network access: an integrated transport layer solution," in *19th IEEE Int. Conf. Network Protocols*, Vancouver, Canada, Oct. 2011, pp. 353-362. doi: 10.1109/ICNP.2011.6089074.
[13] U.S. Department of Transportation. (2014, Jan.). *Vehicle-to-Infrastructure (V2I) Communications for Safety* [Online]. Available: http://www.its.dot.gov/research/v2i.htm
[14] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using WiFi," in *Proc. 14th ACM Int. Conf. Mobile Computing and Networking*, San Francisco, CA, USA, Sept. 2008, pp. 199-210. doi: 10.1145/1409944.1409968.
[15] W. Shang, J. Thompson, M. Cherkaoui, J. Burke, and L. Zhang, "NDN.JS: a JavaScript client library for named data networking," in *Proc. IEEE INFOCOMM 2013 NOMEN Workshop*, Turin, Italy, Apr. 2013.
[16] D. Kulinski and J. Burke, "NDN video: live and prerecorded streaming over NDN," NDN Project Team, Tec. Rep., 2012.
[17] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, (2013, Oct. 22). *Vehicular Inter-Networking via Named Data* [Online]. Available: http://arxiv.org/pdf/1310.5980.pdf
[18] NHTSA. (2014, Feb. 3). *U.S. Department of Transportation Announces Decision to Move Forward with Vehicle-to-Vehicle Communication Technology for Light Vehicles* [Online]. Available: http://www.nhtsa.gov/About+NHTSA/Press+Releases/2014/USDOT+to+Move+Forward+with+Vehicle-to-Vehicle+Communication+Technology+for+Light+Vehicles

Manuscript received: February 25, 2014

Biographies

Mario Gerla (gerla@cs.ucla.edu) obtained his undergraduate engineering degree from Politecnico di Milano, Italy. He received his PhD degree from UCLA. In 2002, he became a fellow of the IEEE. As a graduate student at UCLA, he was part of the team that worked on the early ARPA Network system and protocols under the guidance of Professor Leonard Kleinrock. After four years at Network Analysis Corporation, New York, he joined UCLA in 1976. At UCLA he designed network protocols, including ad hoc wireless clustering, multicast (ODMRP and CODECast) and Internet transport (TCP Westwood). He has lead the ONR Mimuteman project, designing the next-generation scalable airborne Internet for tactical and homeland defense scenarios. He is now leading several advanced wireless network projects funded by industry and government. His team is developing a vehicular testbed for safe navigation, content distribution, urban sensing and intelligent transport. Parallel research activities are wireless medical monitoring using smart phones and cognitive radios in urban environments. He has served on several conference program committees, including MobiCom, MobiHoc, MedHocNet and WONS. He is on the IEEE TON scientific advisory board.

Giovanni Pau (giovanni.pau@lip6.fr) is the ATOS/Renault smart mobility chair professor at the University Pierre at Marie Curie, Paris. He received the Italian Laura in computer science in 1998. He received his PhD degree in computer engineering from the University of Bologna in 2002. Before Joining UPMC, Dr. Pau was a senior research scientist at the UCLA Computer Science Department, where he is currently an adjunct professor.
Dr. Pau's core research interests include network systems with a focus on vehicular networks and pervasive mobile sensor systems. He designed and built the UCLA campus vehicular testbed and the UCLA/MPI urban sensing testbed designed to enable hands-on studies on vehicular communications and urban sensing. His research contributions lead to the VERGILIUS and CORNER simulation suites designed to support mobility and propagation modeling in urban environments. More recently, Dr. Pau designed and developed VNDN the Named Data Network (NDN) protocol stack specifically adapted to work on mobile-to-mobile scenarios.

Rita Tse (ritatse@ipm.edu.mo) is the program coordinator in the Computing Program at the Macao Polytechnic Institute. Her primary research interests are on ubiquitous computing and urban sensing. She published several papers in esteemed international journals and conferences. She is currently serving as local manager for the UCLA-MPI Joint Research Laboratory in Ubiquitous Computing.

DOI: 10.3939/j.issn.1673-5188.2014.01.007
http://www.cnki.net/kcms/detail/34.1294.TN.20131211.1242.001.html, published online December 11, 2013

Mobile Internet WebRTC and Related Technologies

Zhenjiang Dong, Congbing Li, Wei Wang, and Da Lyu
(ZTE Corporation, Shenzhen 518057, China)

Abstract

This paper describes an improved design for WebRTC technology. With this design, WebRTC communication at client side, server side, and between these two sides is improved. HTML5 WebSocket, media negotiation and synthesis, network address translator (NAT)/firewall traversal, Session Initiation Protocol (SIP) signaling interaction, and P2P communication security are all used in this improved design. This solution solves cross-browser running problem of WebRTC applications, reduces reliance on client-side processing capability, and reduces bandwidth consumption. With this design, WebRTC also become more scalable.

Keywords

WebRTC technology; application model; running across browsers; extension

1 Introduction

Web Real-Time Communication (WebRTC) [1] is a real-time audio and video communication technology based on Web browsers. It enables developers to use simple Web technologies to quickly and easily develop rich, real-time online multimedia applications without installing any plug-ins.

As network bandwidth and Web browsers continue to improve, WebRTC will have a huge impact on traditional real-time communication, and it is gradually becoming a mainstream real-time communication technology. Google has provided WebRTC support in its new version of Chrome on the Android platform. It is predicted that at least one billion terminals will support WebRTC in 2013. In an investigation results by CTI [2], a converged communications forum in China, 87% of the telecom companies in the survey are considering making WebRTC part of their product strategies; 86.9% of these companies say that WebRTC is a significant part of their overall product strategies; and 49% of these companies intend to deploy WebRTC solutions within a year.

WebRTC is developed by Global IP Solution Corporation, which was acquired by Google in 2010. After this acquisition, Google decided to open WebRTC technology to the public. Both the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) have formed WebRTC standardization groups, and WebRTC standards are being supported by more and more manufacturers within the industry.

Compared with traditional communication technologies, WebRTC has the following advantages:

- It is easy to use, and plug-ins or different platform client applications do not need to be installed.
- It provides a consistent user experience.
- It can be quickly and easily upgraded at the server side.
- On the basis of WebRTC, web instant communication services can be developed with JavaScript or HTML.
- It can run across operating systems, which means that developers do not need to develop different application versions for different operating system.
- Developers can focus on services rather than media processing.

2 WebRTC Standard Progress

W3C and IETF are responsible for developing WebRTC standards, but these groups have a different target.

The W3C WebRTC working group aims at defining client-side Javascript APIs. With these APIs, Web applications can complete point-to-point or point-to-multipoint real-time communication between browsers. Key members of the W3C WebRTC working group are Web browser vendors, such as Google, Mozilla, and Opera.

The IETF RTC-Web working group defines real-time communication protocols and media formats between browsers. That is, it focuses on media codecs, network transmission, Network Address Translator (NAT)/firewall traversal, security, privacy, etc. Key members of IETF RTC-Web are Microsoft, Google, Skype, Yahoo!, Cisco, and FT (Orange).

Initially, WebRTC standards have three types of APIs: NetWork Stream, RTCPeerConnection, and DataChannel. NetWork Stream API is used for user media acquisition; RTCPeerConnection API is used to establish connectivity between browsers; and DataChannel API is used to transmit user data except video and audio stream from camera or microphone. Both NetWork Stream API and RTCPeerConnection API were developed earlier than DataChannel API and are used more widely. NetWork Stream API is independent from WebRTC standard in recent version.

As WebRTC standards mature, they are being supported by more and more modern browsers (**Table 1**).

This work is supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China (2011ZX0300200201).

▼**Table 1. Browsers that support WebRTC**

Developer	Browser that Supports WebRTC
Google	Chrome 23+ for desktop, Chrome beta for Android
Mozilla	Firefox 22 beta+
Opera	Opera 12+

To verify the effectiveness of WebRTC in Web instant communication, browser vendors Google and Mozilla and third parties, such as Mobicents [3] (a VoIP middleware platform provider acquired by Red Hat), Livecome [4] (a communication service solution provider), and some Web developers, have developed their own WebRTC demos. Each of these WebRTC demos can be experienced with a camera and a microphone in WebRTC-supported browsers.

3 WebRTC Application Modes

3.1 Adding WebRTC Access to Traditional Audio and Video Services

Telecom operators and solution providers treat WebRTC as a new access mode for their traditional audio and video services. For example, users access a traditional call center based on web browser, and the browser becomes a new terminal of a traditional conference system.

The difficulty of this mode is ensuring compatibility between the WebRTC and traditional application architecture. A gateway device is added between WebRTC and the traditional application architecture in order to coordinate differences between them. For example, in ZTE's WebRTC-IMS gateway access solution, the WebRTC-IMS gateway is responsible for signaling conversion, media streaming relay, and NAT/firewall traversal mechanism conversion between WebRTC and traditional IMS networks.

The advantages of this solution are:

- A browser access mode is added in traditional audio/video service and no browser-side plug-ins need to be developed. This will greatly reduces client-side work.
- Media processing capabilities of WebRTC are strengthened, and user experience is improved by reusing Media Server in traditional architecture.
- Dependency on client-side processing capability is reduced and network bandwidth consumption are reduced when using WebRTC access traditional service.

3.2 Lightweight Real-Time Web Communication

The typical application scenario of WebRTC is to use WebRTC standard JavaScript APIs to develop lightweight Web audio/video calls and Web audio/video conference centers. Representative demos within the industry field are Mobicents SIP Servlet [5], Conversat.io[6] (now renamed as Talky.io), and Chatdemo [7],etc.

In this application scenario, almost all demos adopta decentralizedidea on the application architecture design. Its key points is decomposingmultiparty media negotiation process into multiple end-to-end negotiation processe, and functions like audio/video processing implemented in traditional media server are transferred to the client browser side.

Once obtained media formats and transmission protocol supported by both parties, two browsers can communicate with each other and media streams can be sent to the other side directly onther than relayed by the server. When a third party joins in, the third-party UA should initiate negotiations with the earlier two parties' UA. After this step ,the following process are the same as communication between two parties.

This solution is very easy and do not need to considering media processing. However, client-side media processing pressure will become increasingly heavier. In short, this solution largely dependend on client-side processing capabilities and therefore can't support more than six users in a video conference.

3.3 Comprehensive Web Real-Time Communication Service

Compared to the above one, the biggest difference is that we add a Multipoint Control Unit (MCU) in the server-side. Unlike the above solution, all media processing are concentrated in the MCU. Besides, MCU also provide some strengthen functions , such as audio/video mixing .

Because media processing are transferred to server-side, this solution significantly reduces WebRTC applications' dependence on client-side processing capability and extends the application range of WebRTC technology.

All in all, the last solution takes the advantages of the former two and can be applied in the large-scale application scenarios.

4 WebRTC Technical Solution: Problems and Improvement

4.1 Technical Problems in Current WebRTC Applications

As a developing technology and standard, WebRTC has many defects that represents a real challenge in its spread process. These defects are as follows:

- Compared with traditional video conferencing service, WebRTC has little control over user media. This characteristic means WebRTC is not adaptable to the requirements of complex conferencing.
- Considering media streams are sent to each other directly in WebRTC, client-side processing capability is very important. Less than five parties in a PC video conference is ok, but user experience suffers when this number number becomes larger. This problem is worse in the mobile terminal.
- Enormous amount of bandwidth consuming will Significant-

ly increase the cost of using WebRTC services.

- There are some differences in WebRTC APIs' name or HTML5 APIs' name in different browsers[8],[9]. WebRTC applications running at one browser successfully may encounter exceptions while running at another browser environment.
- WebRTC APIs can operate local media files and devices , this access mode may results in potential threats to the system [10].

4.2 Suggested Improvement

We propose an improvement solution in use of WebRTC to develop lightweight real-time Web communication services. This solution resolves the cross-browser running problem, reduces the dependencies of client-side processing capabilities and bandwidth consumption, and improves system scalability (**Fig. 1**).

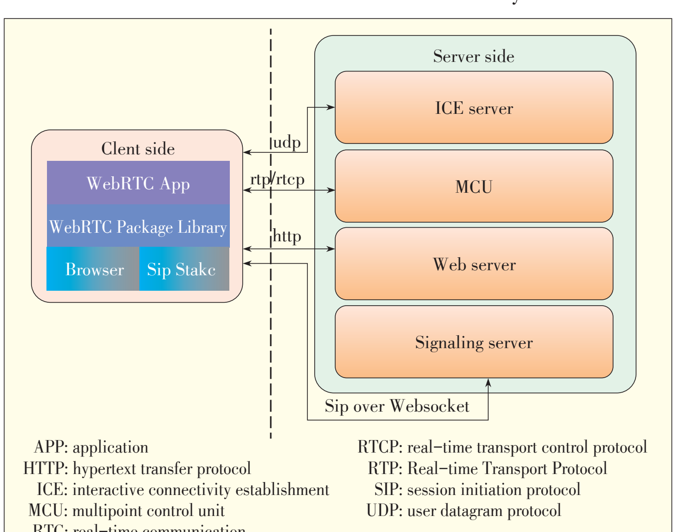
Fig. 1 shows the client side network elements (NEs), server side NEs, and how to communicate between these two sides.

4.2.1 Client Side

On the client side, the browser must support WebRTC standard interfaces, and the SIP Stack supports SIP signal processing. The WebRTC package library is based on WebRTC Javascript APIs and SIP Stack. Its implements: two-way communication between the client side and server side, media and firewall traversal negotiations between browsers, and running cross-browser support. The WebRTC APP completes the user interface display [11].

4.2.2 Server Side

The server side includes interactive connectivity establishment (ICE) [12] server, MCU, Web server, and signaling server. The ICE server interacts with the client-side browsers and returns iceCondidates to the client. The MCU synthesizes video streams transmitted from the browsers of various parties and sends the final stream back tothose browsers concerned in the session. The Web server provide the running environment for the WebRTC APP. The signaling server processes, routes, and distributes SIP requests from the client side and implements user authentication, user management, conference controlling and recording, and conference charging, etc. In an actual deployment scenario, the above four network elements can either be separately deployed or deployed on the same server or several servers.



▲**Figure 1. System architecture of a lightweight Web real-time communication service.**

4.2.3 Communication Between Client Side and Server Side

Communication between ICE client and ICE server complies with Simple Traversal of UDP over NATs (STUN) [13] and Traversal Using Relays around NAT (TURN) [14] protocols. UDP protocol is used for message transmission. SIP over WebSocket [15] technology is used for two-way SIP interaction between the client side and server side. Real-Time Transport Protocol/Real-Time Transport Control Protocol (RTP/RTCP) are used for transmission of the media stream between the client side and server side. HTTP is used for access to Web services.

Within this framework, a general multiparty video communication procedure can be described in the following steps:

1) Users access services provided by the Web server through client-side browsers.
2) Client browsers are registered to the signaling server and implement multiparty point-to-multipoint media negotiation. The MCU receives local audio and video streams transmitted by each party in the conference in the given format. It performs sound mixing, frequency mixing for these streams, then returns the synthesized stream to each party.
3) The local browser of each party receives and displays the media stream.
4) A multi-party communication process is established.

4.3 Key Technologies

4.3.1 HTML5 WebSocket

Traditional real-time communication is usually use polling or long-polling technology. Through increasing the frequency of client requests or extending server response time, developers can improve the user experience of real-time service to some extent, but an HTTP-based request-response mode cannot provide true real-time communication. In addition, polling or long polling technology increases client-side bandwidth consumption and server-side resource consumption.

To reduce delay in real-time communication and decrease bandwidth consumption, the HTML5 working group has developed a WebSocket communication standard. HTML5 WebSocket defines a full-duplex communication channel through a single socket on the Web and thus significantly improves real-time Web communication.

Before the client side and server side can communicate with each other, they need to complete a WebSocket handshake to establish a connection between them. Once the connection has been established, data can be transmitted between the two sides in both directions. In this way, when the server side updates data, this data can be directly pushed to the client side, and there is no need to wait for a request from the client side.

In very strong real-time applications, such as audio and video communication, HTML5 WebSocket can greatly improve performance and reduce bandwidth consumption by reducing transmission payload.

4.3.2 Media Negotiation and Synthesis

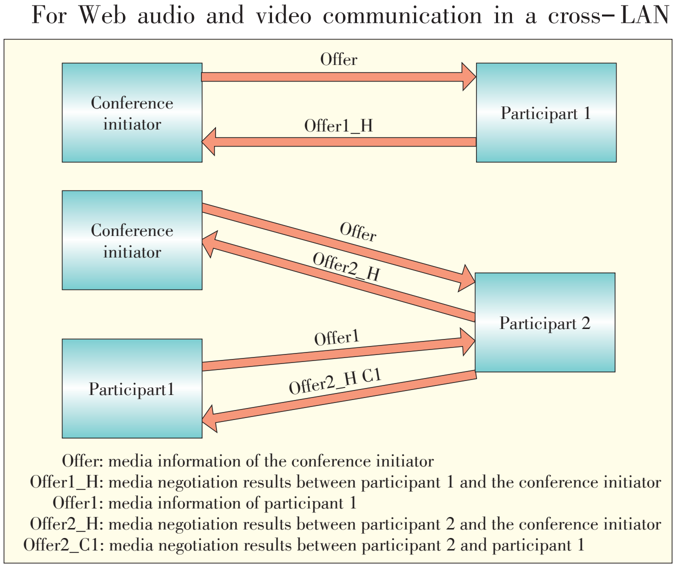
In this improvement, the media negotiation still occurs on the client sides. When there are multiple parties in a conference, the multiparty negotiation process needs to be decomposed into multiple end-to-end negotiation processes. Optional negotiation strategies are shown in **Fig. 2** and **Fig. 3**.

Theoretically, after a multiparty media negotiation has ended, multiparty media streams can be directly communicated to each other. However, this application mode creates significant issues. In a four-party conference, for example, each party needs to send out three-way local video streams at the same time and receive video streams from the other three parties. In this way, the client side can only support a very limited number of users, and a huge amount of network bandwidth is consumed.

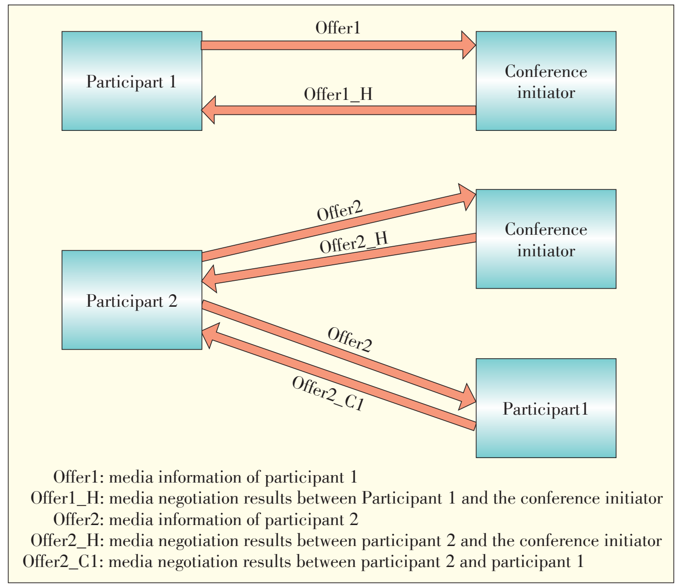
Therefore, an MCU is needed on the server side for media stream conversion, synthesis, and so. After the MCU has been introduced in this solution, each client only needs to send out one-way local video streams and receive one-way synthesized video streams from the MCU.

4.3.3 NAT/Firewall Traversal

For Web audio and video communication in a cross-LAN environment, NAT/firewall traversal equipment is used to route the transmission of media streams. The IETF-RTC Web standards stipulate that WebRTC browsers needs to support the ICE framework and the server side needs to provide the corresponding ICE server.



▲Figure 2. Negotiation process initiated by earlier entrants.



▲Figure 3. Negotiation process initiated by later entrants.

As for the selection of traversal schemes, STUN and TURN servers are supported at the same time. The STUN server completes non-symmetric NAT traversal, and the TURN server completes symmetric NAT traversal and firewall traversal. Through the synergy of the STUN and TURN servers, WebRTC media stream transmission is guaranteed.

4.3.4 Signaling Interaction Technology Based on SIP Stack

In the W3C WebRTC standard, signaling between the client side and server side is not standardized. SIP is a simple, flexible, scalable protocol and become more and more popular in the industry filed. It should be said that SIP has become the virtual standard of next-generation communication.

To complete SIP signaling interaction between the client side and server side, we adopt the reference implementation of Mobicents JAIN-SIP on the client-side, and a B2BUA is developed on the server-side to process and distribute the requests from the client-side. **Fig. 4** shows the process for a server to act as the proxy and do routing distribution.

4.3.5 P2P Communication Security

To prevent browser programs from abusing local resources, we refer to [10] and introduce a sandbox into the WebRTC framework. The core idea of sandbox technology is to classify external websites as either trusted or untrusted. Trusted websites are considered safe and can access all local resources (or a limited number of the resources), and untrusted websites are put into the sandbox and cannot directly access local resources.

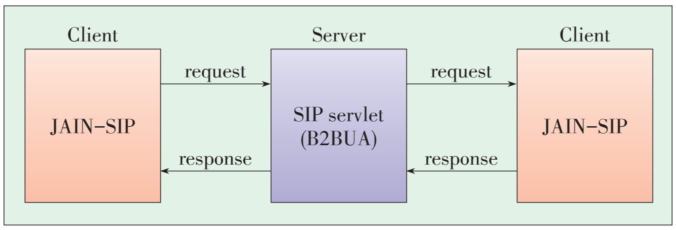


Figure 4. A server acting as a proxy in order to perform routing distribution.

To filter the trusted and untrusted websites, the Same-Origin Policy (SOP) [16] of the IETF is used. That is, the security of a webpage is determined by its origin, and it is necessary to perform homologous match on the protocol, host, and port that the webpage uses. A matching security access policy is assigned to each origin. For example, when a user accesses resources on a website B from website A, then the user is not allowed to access the encrypted local files on website B.

To prevent malicious websites from launching Distributed Denial of Service (DDOS) [17] attacks through Javascript API or simulating a domain name server (DNS) through a browser protocol packet and attacking a company's DNS, auxiliary defensive measures are introduced. These measures include regular scanning and checking the origins of visitors.

5 Enhancement and Optimization

To enhance efficiency and user experience in a lightweight WebRTC communication system, the following need to be optimized:

- video stream switching. In a traditional video-conferencing system, the video of a participant may be enlarged, and the screen may presents one big user interface and several smaller ones. In a WebRTC system, the <video> label of HTML5 needs to be used. A video stream is switched on demand by dynamically replacing the src attribute of the <video> label. Dynamically changing the src attribute is difficult because, unlike static video stream switching, the src attribute of each <video> label cannot be assigned in advance. In a WebRTC system, participants are dynamically added, and video streams are also dynamically obtained. To achieve video stream switching, video streams of the participants need to be stored, and the video streams have to correspond with their respective video container IDs in advance. When a user clicks a video, the application program determines which video stream needs to be enlarged and switched according to the video container ID.
- local echo elimination. In some WebRTC demo systems, media streams obtained from peripherals (such as cameras or microphones) are directly added to the local browser. This means that all participants can hear a local microphone echo before joining the conference. To eliminate this effect, two copies of the local media streams need to be saved before they are added to the browsers. Video streams from one of the copies (i.e. eliminating audio streams) needs to be extracted in order to add them to the local browsers. Then, the other complete copies of the media streams (including video and audio streams) are sent to the other participants through WebRTC interfaces. In this way, the other parties can hear the local voices, and the local audio echoes are eliminated.
- access support for non-WebRTC terminals. Such terminals include existing ZTE conference terminals, Polycom, Microsoft Lync, and IBM Sametime. On the one hand, various control protocols are supported; on the other hand, the system side needs to complete the conversion of video and audio streams.
- cross-browser support. The best browser that supports WebRTC is Google Chrome. Firefox22 comes in second place, and other browsers have weak support for WebRTC. There are interoperability problems between different browsers and currently, WebRTC is often restricted to a single browser. In order to run across browsers, we need to: 1) shield the differences between different browsers on the WebRTC interfaces in the WebRTC package library (including RTCPeerConnection, RTCSessionDescription, and RTCIceCandidate); 2) shield the differences between different browsers on implementation of HTML5 interfaces (including URL, MediaStream, and GetUserMedia) in the WebApp layer; and 3) shield the differences between different browsers on a webpage layout on the CSS layer.
- system-side flow reduction and linear expansion. Depending on the application scenario and access terminals, the system determines (automatically or according to policies) whether data streams between terminals are sent directly or sent through the central server. In a two-party or three-party scenario, if the terminals are the same type (video transcoding by the center is not required), they can be directly connected to each other, and data streams do not need to be forwarded through the center. This significantly reduces the pressure on the system side.
- integration with other systems, such as instant messaging (IM) systems.
- open system capabilities, open APIs (such as Rest), and support for third-party development.

6 Conclusion

WebRTC technology and standards are not mature yet, and there are still many problems in WebRTC implementation. For example, many functions are still not defined, and different browsers are incompatible with each other. There are also strict requirements on for client-side processing capability, and bandwidth consumption is high. This paper describes an improvement for WebRTC real-time communication. This improvement solves the main problems encountered in the applicationof WebRTC and is important to the development of WebRTC technology.

With the popularization of LTE, Wi-Fi and wired broadband; the extension of bandwidths; and the increased processing capabilities of PCs and mobile terminals, browsers on various terminals will support WebRTC so that real-time audio and video communication will become universal. Comprehensive Web communication is an ideal application mode.

HTML5 and WebRTC technologies and standards are maturing and spreading rapidly. Traditional IM,desktop sharing, electronic whiteboard, and audio and video conferencing will gradually be Web-based and will eventually be combined into a complete, unified communication system. Users will be able to communicate through browsers, and interpersonal interaction will become simpler and faster.

References

[1] *WebRTC 1.0: Real-time Communication Between Browsers* [Online]. Available: http://www.w3.org/TR/Webrtc/#rtcpeerconnection
[2] *Dialogic: 87% telecom leaders consider to put WebRTC into product strategy* [Online].Available: http://www.ctiforum.com/news/baogao/370610.html
[3] *Mobicents Project* [Online]. Available: http://hudson.jboss.org/hudson/view/All/job/Mobicents/
[4] *livecome Webrtc demo* [Online]. Available: http://lwork.hk:8086
[5] *Mobicents SIP Servlet* [Online]. Available: http://dev.telestax.com/sipservlets/wiki/HTML5WebRTCVideoApplication
[6] *talky.io* [Online]. Available: http://talky.io
[7] *chatdemo* [Online]. Available: http://ishare.iask.sina.com.cn/f/35083616.html
[8] *simpleWebrtc* [Online]. Available: https://github.com/HenrikJoreteg/SimpleWebRTC
[9] *Webrtc.io* [Online]. Available: https://github.com/WebRTC/Webrtc.io- client/blob/master/lib/ Webrtc.io.js.
[10] L. Li. "Security Analysis of Web Browser-Based P2P Real-Time Communication," *NetWork & Computer Security*, vol. 06, pp. 54-56, 2012.
[11] *Session Initiation Protocol* [Online]. Available: http://www.rfc- editor.org/rfc/rfc3261.txt
[12] *RFC5245-Interactive Connectivity Establishment* [Online]. Available: http://www.packetizer.com /rfc/rfc5245/
[13] *RFC3489-Simple Traversal of User Datagram Protocol* [Online]. Available: http:// www.ietf.org/rfc/rfc3489.txt
[14] *RFC5766-Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)* [Online]. Available: http://www.rfc-editor.org/info/rfc5766.
[15] *draft-ietf-sipcore-sip-Websocket-04* [Online]. Available: http://tools.ietf.org/html/draft-ietf-sipcore -sip-Websocket-04.
[16] *IETF-Security Considerations for RTC-Web draft-ietf-rtcWeb-security-02* [Online]. Available: https://datatracker.ietf.org/doc/draft-ietfrtcWeb-security.
[17] M. Wu, R.C. Wang, and Z.P. Wang. "Detection mechanism of DoS attacks in Peer-to-Peer networks based on support vector machine," *Computer Technology and Development*, vol. 19, no. 11, pp: 151-154, 2009.

Manuscript received: September 25, 2013

Biographies

Zhenjiang Dong (dong.zhenjiang@zte.com.cn) received his MS degree from Harbin Institute of Technology. He is the leader of Business Expert Team of ZTE Expert Committee for Strategy and Technology and the deputy president of ZTE Cloud Computing and IT Research Institute. His research interests include cloud computing, big data, new media, mobile Internet, and other technologies. He presides over 10 funded programs and has published more than 10 academic papers and one monograph.

Congbing Li (li.congbing@zte.com.cn) received his MS degree from Nanjing University of Science and Technology. He is currently a technical researcher in the field of mobile Internet at ZTE Service Research Institute. His research interests include WebRTC, node.js, HTML5, open platform, and other technologies. He is responsible for researching mobile Internet IM services and implementing related demos.

Wei Wang (wang.wei8@zte.com.cn) received her MS degree from Nanjing University of Aeronautics and Astronautics. She is a ZTE engineer and project manager in the field of mobile Internet at ZTE Cloud Computing and IT Research Institute. Her research interests include new mobile Internet services and applications, PaaS, terminal application development, and other technologies. She has authored five academic papers.

Da Lyu (lvda@zte.com.cn) received his MS degree from Nanjing University. He is deputy president of ZTE Cloud Computing and IT Research Institute. His research interests include Stored Program Control (SPC) exchanges, IP-based video solutions, and value-added services.

ZTE Launches the World's First Converged Intelligent Videoconferencing Solution at ISE 2014

February 4, 2014—ZTE Corporation has launched the world's first converged intelligent videoconferencing solution at Integrated Systems Europe, Amsterdam. The solution transforms existing video conferencing terminals from functional machines into smart machines.

Due to restrictions in system convergence, traditional videoconferencing can no longer meet the requirements of rapidly developing technologies and markets. The development of video codec technology and the extensive use of wireless technologies such as LTE are also raising people's expectations for higher video resolution and more flexible video communications. This converged intelligent videoconferencing solution uses an open video communication interface, provides a universal platform, and supports convergence with third-party applications, quickly meeting different user requirements. The system has pre-installed common conferencing software and supports plug and play USB flash disks and portable hard disks for easy sharing of electronic documents. With integrated functions such as E-whiteboard, data conferencing and multi-touch, the system supports simple and intuitive operations. (ZTE Corporation)

DOI: 10.3969/j. issn. 1673-5188. 2014. 01. 008
http://www.cnki.net/kcms/detail/34.1294.TN.20131123.1705.001.html, published online November 23, 2013

Design and Implementation of a Distributed Complex-Event Processing Engine

Ping Lu, Yuming Qian, and Kezhi Zhu
(Cloud Computing & IT Institute, ZTE Corporation, Nanjing 210012, China)

Abstract

A high-performance, distributed, complex-event processing engine with improved scalability is proposed. In this new engine, the stateless processing node is combined with distributed storage so that scale and performance can be linearly expanded. This design prevents single node failure and makes the system highly reliable.

Keywords

CEP; stream processing; M2M; sliding window; real-time processing

1 Introduction

The development of mobile internet and the internet of things (IoT) has caused a data explosion that has been the impetus for continuous improvements in data processing. Ten years ago, diversified enterprise systems were isolated information islands. Since then, service-oriented architectures (SOA) have been widely researched in terms of acquiring information and connecting information islands. However, quickly acquiring valuable content from mass data and preventing useless information from swamping valuable content is an issue.

In telecommunication applications, user signaling and behavior is monitored in real time. Any internet access request, call, short message, or location change requires real-time data collection and processing. At the same time, relevant behavior models also need to be built. The amount of data processing becomes so huge that in a city of two million people, the amount of signaling reaches a level of gigabytes per second. This cannot be handled with traditional offline processing.

There are two main solutions to processing mass real-time data: online data collection and offline data processing (using a method similar to MapReduce) and event streaming, in which data is calculated and processed in the memory. Micro-MapReduce is an example of the former for an event system [1]. Micro-MapReduce can reduce the granularity and shorten the period of MapReduce. It also properly addresses scalability (although near-real-time extends from five minutes to an hour). Event streaming includes current open-source stream-processing frameworks, e.g. S4. Oracle CEP is an available S4 product [2]. With event streaming, relevant data can be loaded and calculated in the memory. Therefore, this method provides high-performance standalone processing, but is weak in terms of scalability and fault and disaster tolerance. Data therefore needs to be distributed at the front end and aggregated at the back end.

Storm is a good distributed solution [3]. A Storm cluster comprises a master node and multiple worker nodes. The master and workers interact through a Zookeeper. Storm is essentially a reliable distributed event-processing engine that ensures every event is processed. However, single-point failure at the Storm nimbus (master) node is a problem, and HA is required. Moreover, without a time window, Storm cannot sufficiently support either event windows or multi-event collaboration.

With complex event processing (CEP), a user-friendly interface is essential. Event Processing Language (EPL), a Quasi-structured Query Language (SQL), is usually used to define event processing logic. Cayuga [4] and Borealis [5] are effective systems for EPL processing and guaranteeing event QoS.

To reliably process mass real-time data, we propose a new high-performance distributed CEP engine called ZX-CEP. This engine has the following features:

- stream computing for complex event data
- high concurrency (a single server supports more than 100,000 messages per second)
- high linear scalability
- simple orchestration of EPL message processing and graphic processing flow
- distributed computing
- linear scalability of system capacity and processing
- sliding event window.

2 Distributed Stream-Computing Architecture

A distributed stream-processing system can be regarded as a black box. Large numbers of continuous data streams enter the black box, where they are processed and converted into specific event streams for output or for further processing in other systems. If the system detects an alarm during stream processing, the system generates an alarm event and requests an automatic maintenance program to fix the fault or else it stores the event in a persistent storage engine for future analysis and processing.

In a stream-processing system, the data stream directions conform to a directed acyclic graph. If data multiprocessing is needed, the output of one flow can be used as the input of another flow. This is serialized processing of multiple flows.

Fig. 1 shows the architecture of the distributed complex-message processing engine, which comprises data-preprocessing module, complex- message processing module, output adapter, and task scheduling management module.

2.1 Data-Preprocessing Module

Continuous event streams are sequences of events that have not undergone structured processing. These original events are filtered, combined, and distributed by the event-preprocessing module (**Fig. 2**).

The preprocessing module is responsible for input data adaptation and preprocessing.

The input data adapter receives original events and converts them to structured events. It also follows the event occurrence sequence and puts the converted events into a local message queue, where they await preprocessing. An input data adapter is usually customized according to the input contents. After input data adaptation, original events are converted to format messages that include message source ID, generation time stamp, and content K/V object. Messages that cannot be quantized require metadata management, which involves quantization mapping of message content.

The following atoms for event preprocessing have been achieved: field filtering atom, field filling atom, event filtering atom, event combination atom, and event split atom. The task manager customizes rules for and dynamically loads basic atom operations that can be instantiated into multiple operators. In data preprocessing, these operators are connected according to defined rules. Operators can be connected using a graphic editing tool or through EPL condition resolution. Connecting operators in a pipeline manner is beneficial because complex data-processing logic can be generated any time by connecting basic operators in series and parallel, and complicated code does not have to be written.

During event processing, an input signal may generate some noise above the normal amplitudes. Filtering removes the noise so that the signal is normal [6].

2.2 Complex Message-Processing Module

Multiple event-processing modules detect one or more window-change event queues. Idle event-processing modules au-

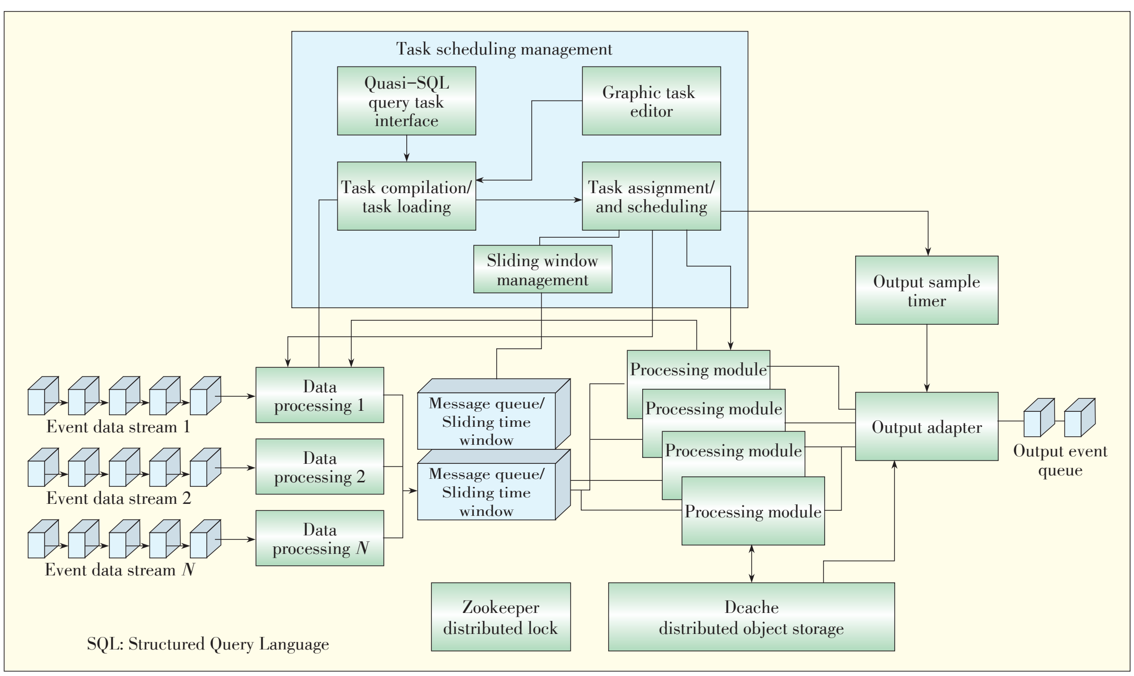


Figure 1. Architecture of the distributed-complex message processing engine.

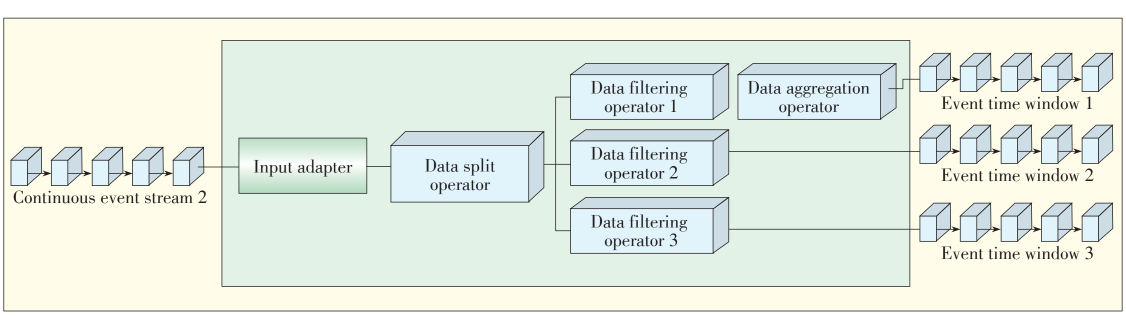


Figure 2. Architecture of the data-preprocessing module.

tomatically acquire events from the queue or queues. Because the event-processing modules are stateless, they can be added or removed any time according to service conditions and without affecting system operation.

In distributed message processing, a distributed message queue ensures that events are sequential. This can be achieved in the DCache K/V system [7] or by using other high-performance distributed message queues. The sequencing and distributed processing of messages is ensured by maintaining a consistent message queue in the distributed storage (**Fig. 3**). In a distributed K/V system, there is also a unified time window that is maintained by the elected master node. This avoids processing errors caused by inconsistency between the clocks of different nodes.

2.3 Output Adapter

The output adapter converts system processing results into specific output actions or data streams. The output includes message output and periodical sampling output. If the output is a data stream, it can act as an input stream that is later processed according to another group of rules. In this case, analysis should be performed in different dimensions. The output adapter observes data stream conditions within the fine granularity of five minutes, and the output of integrated results forms a data stream with a coarse granularity. Then, analysis is done over a longer period, such as one day, and the outputs are files, database tables, and message queues. The output adapter is generally customized according to service requirements, and it also supports time travel. After a rule condition has been triggered, the output adapter can record relevant system information before and after the event has occurred. The records are put into mirror persistent storage for later replay and analysis.

2.4 Task Scheduling Management Module

The task scheduling management module includes rule generation and rule scheduling and execution (**Fig. 4**).

Rule generation involves dynamic generation of customized task graphs by EPL event processing statements. It also involves the generation of event processing logic in GUI mode by a rule editor.

Rule scheduling and execution involves dynamic rule loading in the service process. That is, the rule loading procedure cannot affect the normal processing procedure. EPL is suitable for simple rule scenarios, and rule graph editing is suitable for complex rule scenarios. A graphic rule editor has been developed to increase efficiency. It converts rule graphs into corresponding codes and dynamically loads the codes.

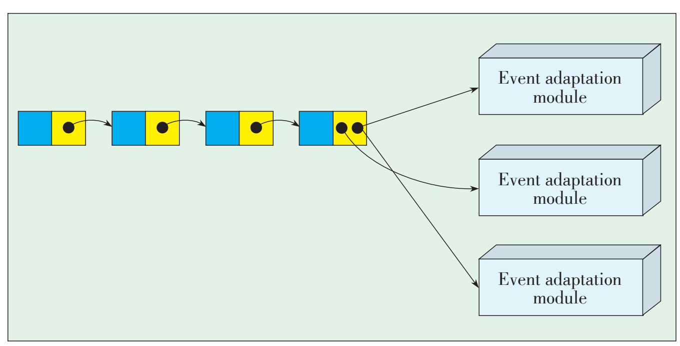
When a designated sliding window has adapted to a new rule, old data (that was generated before the new rule took effect) has to be matched. We propose two strategies for addressing this problem. First, after the new rule has been deployed, the corresponding window data is cleared. However, this may cause data interruption. Second, the time can be recorded after the new rule takes effect, and the old and new rules can be simultaneously computed for the recorded period. Computing results obtained using the new rule are not used until the data generated after the new rule takes effect has been removed from the stack. Otherwise, computing results obtained by using the old rule are always in use. A possible effect of this is that only after a time $T = T_w + 1$, where W is the time window length, can the new rule take effect.

3 Sliding Window Design and Dirty Data Taint Propagation Mechanism

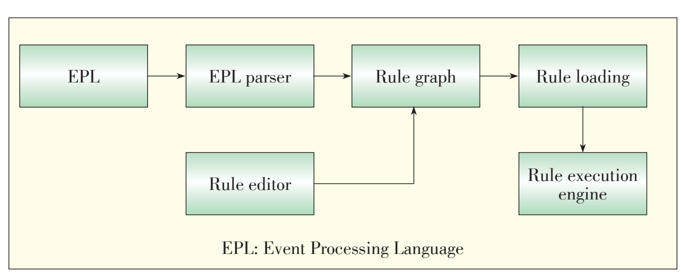
The sliding event window can be achieved through a high-performance distributed message queue. This window is defined as a unique event sequence with a fixed time length or a fixed number of messages kept in the system. As time passes, all messages kept in the event sequence stay within the specific time or length range. Therefore, there are sliding time windows and sliding space windows. In a sliding time window, all events are maintained within a specific time interval. In a sliding space window, event capacity is defined in advance, and events exceeding this capacity are automatically removed from the stack.

The distributed K/V engine maintains the sliding event window so that storage of events is guaranteed and consistent.

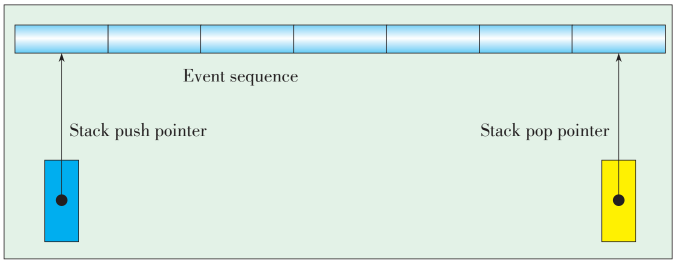
Each element of the event sequence, stack push pointer, and stack pop pointer are kept in the distributed K/V engine as key-value pairs. In this way, the distributed sliding event window is stored (**Fig. 5**). The stack push pointer and stack pop pointer are accessed in a specific synchronous operation mode,



▲**Figure 3. Multiple event processing modules' distributed processing of the same change queue.**



▲**Figure 4. Rule generation and loading.**



▲Figure 5. Distributed sliding event window.

which ensures data consistency in a distributed environment.

A message processing action is triggered by either the entry of an event into the event window or the exit of an event from the event window. Such entries or exits are detected by periodic scanning, and an action triggers the complex message processing module. The message queue mechanism guarantees distributed processing and implements message processing in pull mode. When an event enters or exits the sliding event window, a change message is generated. This message enters another message queue and waits for the complex message processing module to respond and handle it.

We build a data model similar to Aurora [9] and convert the event sequence of the time window into an incremental event sequence. There are three types of incremental events:

1) insertion event, given by (+, t), where t is an event object newly added to the event window
2) deletion event, (−, t), where t is an event object that exits the event window
3) replacement event, (^, t_1, t_2), where t_1 is a replaced event object, and t_2 is a new event object.

By handling incremental events, the system avoids frequent global scanning of the event window, and this greatly accelerates the processing. In addition, QoS marks are added to event information and sent to queues with different priorities. In this way, the system prioritizes the processing of events with a high priority. Distributed K/V storage maintains the event state machine and global counter and simplifies the data-processing logic in the event-processing procedure. Here, we discuss computing the average value of all the events in the event window as an example. Typically, the averages of all events in the event window are recalculated using

$$T(k)=\left(\sum_{k=0}^{n}E(k)\right)/n \tag{1}$$

After incremental events have been introduced, the following computation is performed:

$$T(k+i)=T(k)+\sum_{m=k}^{k+1}E(m)/i \tag{2}$$

If the sampling period is 1 s and the event window is 5 min, the amount of computation required using (2) is only 1/300th that required using (1).

However, distributed computing creates a problem: Because multiple event sequences may be involved, events generated by multiple event queues may not be processed by the same event processor. Therefore, we introduce a taint data prorogation model for distributed computing. This model ensures that the updates of any basic event could trigger all necessary subsequent nodes' further processing with in timeframe.

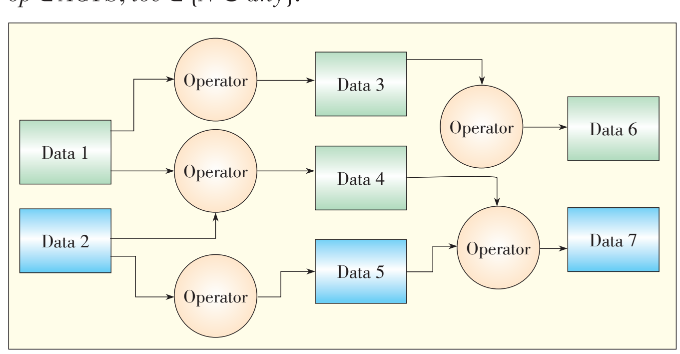
If a rule needs to use data in two or more sliding windows, the event streams generated by two sliding windows may not be analyzed and processed on the same node. Therefore, we propose a distributed taint data propagation mechanism that ensures all the processing nodes of a rule tree correctly update the final results, even if computation is not done on the same node [5]. In **Fig. 6**, gray parts represent dirty data, and dirty data marks are transferred through the data-propagation mechanism. This guarantees all the data is timely updated.

The entire rule tree can be regarded as a directed acyclic graph. After going through an operator, changed data may or may not affect subsequent processing nodes. Dynamic analysis can be done to search out all affected nodes, but static analysis only confirms the trusted boundary. Data outside the trusted boundary may be confirmed as tainted or suspected tainted. Data inside the trusted boundary is guaranteed to be clean.

The taint propagation algorithm recognizes the affected nodes, and then subsequent data of the affected nodes is recomputed by operators. Data that is not affected does not need to be recomputed. In a distributed system, data and operators may be generated or kept in various physical nodes.

Taint propagation analysis can be static or dynamic. The proposed distributed stream computing system does basic static analysis. When source data changes, the system analyzes all affected nodes in subsequent parts of the directed graph and marks all affected data. If affected data needs to be acquired for computing, backward-pass computing can be done according to taint marks. In this way, computation is minimized for the entire system.

The most important part of a taint-check strategy is analyzing the trusted boundary. The taint-check strategy is represented by a tetrad comprising entity type (type), vulnerability description (vul), program operation (op), and operand location (loc): $[type, vul, op, loc] \mid type \in ROLES, vul \in VUL\_TYPES, op \in ACTS, loc \in \{N \cup any\}$.



▲Figure 6. Dynamic data operation of taint propagation.

Each computing node is taint-checked for each input variable, and this helps derive the taint propagation matrix for the entire system (**Table 1**).

In an environment with multiple input variables, confirmed and suspected tainted nodes are combinations of single input variables. The taint propagation algorithm allows the system to update nodes with dirty data only when data output is needed. It is unnecessary to update all data nodes all the time. In this way, system computation is greatly reduced.

▼Table 1. System taint propagation matrix

Input variable	Confirmed tainted node	Suspected tainted node
A	T_1, T_2	V_1
B	T_2	V_1, V_2

4 QoS Guarantee for Event Processing

Messages with different priorities are put into different incremental event message queues according to different QoS marks. In this way, high-priority events are processed first.

The event processing module prioritizes the acquisition of change messages from the high-priority queue. When there are no waiting messages in this queue, the module acquires messages from the low-priority queue.

Computing results of the event-processing module are temporarily stored in DCache or are transferred to the output queue. A sampling tool periodically samples the computing results temporarily stored in DCache and transfers them to the output queue. For example, it may be necessary to acquire the average of a sensor group in the window over the past hour, give a report every five minutes, and send an immediate alarm if the average is too high at a certain time. In this case, the event-processing module modifies the average value object maintained in DCache for each new input or output event. The sampling program samples this average data from DCache at five minute intervals and transfers the data to the output queue. If the data gives ultra-high readings, the program immediately generates an alarm and puts it in the output queue.

5 Conclusion

We propose a distributed complex-message processing engine called ZX-CEP. This high-performance engine can process a complex data stream in real time. The architecture is based on the principle that data should be separated from logic. The data is stored in the cloud, and multiple copies are kept. Data processing nodes are stateless and can be expanded in a distributed, dynamic manner. The engine ensures that storage is scalable and data is secure in a mass-data environment. It also ensures that processing is scalable in the case of parallel processing. Failure of any node does not affect normal processing or stream processing in the system.

This architecture depends on distributed K/V storage and a distributed message queue built on it. A sliding time window is also implemented by using the distributed message queue.

This paper describes how to use EPL to customize and load data-processing logic in real time. EPL is used to complete complex logic orchestration graphs based on basic operators. Distributed data processing and storage involve complex distributed logic computation. A dirty data propagation mechanism is introduced so that data itself drives processing logic.

In the future, the dynamic logic-processing mechanism for this system will be improved so that logical decisions are more flexible and more complex logic computation is supported. Moreover, data storage and calculation nodes will also be adjusted to ensure automatic fault discovery and recovery.

References

[1] T. Condie, P. Alvaro, J. M. Hellerstein, et al., "MapReduce online," Univ. of California, Berkeley, CA, Rep. UCB/EECS-2009-136, 2009.

[2] Qiang Huang and Qingkai Zeng, "Taint propagation analysis and dynamic verification with information flow policy," *Journal of Software*, vol. 22, no. 9, pp. 2036—2048, 2011.

[3] Xinyu Li and Zhongdong Huang, "Persistent event processing solution based on CEP," *Computer Applications and Software*, vol. 27, no. 12, pp. 151-153, 2010. doi: CNKI:SUN:JYRJ.0.2010- 12-049.

[4] M. Cherniack, H. Balakrishnan, M. Balazinska, et al., "Scalable distributed stream processing," in *1st Biennial Conf. Innovative Data Systems Research*, Asilomar, CA, 2003, p. 12. doi: 10.1.1.63.3117.

[5] D. J. Abadi, Y. Ahmad, M. Balazinska, et al., "The design of the borealis stream processing engine," in *2nd Biennial Conf. Innovative Data Systems Research*, Asilomar, CA, 2005, p. 13. doi: 10.1.1.118.7039.

[6] D. C. Luckham and B. Frasca, "Complex event processing in distributed systems," Stanford University, Stanford, CA, Rep. CSL-TR-98-754, 1998. doi: 10.1.1.56.876.

[7] L. Brenna, A. Demers, J. Gehrke, et al., "Cayuga: A high-performance event processing engine," in *ACM SIGMOD Int. Conf. Management of Data*, Beijing, China, 2007, pp. 1100-1102. doi: 10.1145/1247480.1247620.

[8] Oracle. (Dec 17, 2008). *Oracle CEP White Paper* [Online]. Available: https://blogs.oracle.com/CEP/resource/cep_coherence.doc

[9] Hong Gao, "Distributed message queue based on P2P network," *Programmer*, no. 6, pp. 102-106, 2012.

Manuscript received: August 10, 2013

Biographies

Ping Lu (lu.ping@zte.com.cn) graduated from Southeast University, China. He heads the ZTE Cloud Computing & IT Institute and is in charge of R&D of service software, multimedia terminals, signaling detection, and ICT products. He is also responsible for researching new internet, cloud computing, and home network services. He led several national key research projects and has published 10 papers. He is a member of the China Computer Federation and CCF TCSC.

Yuming Qian (qian.yuming@zte.com.cn) graduated from Southeast University, China. He is a chief engineer at ZTE Cloud Computing & IT Institute and is responsible for researching system architecture and new technologies for mass-data processing, cloud computing, and mobile internet. He has led several national key research projects and has published 16 papers. He was awarded the prestigious title Innovation & Entrepreneurship Talent of Jiangsu Province.

Zhike Zhu (zhu.zhuke@zte.com.cn) graduated from Southeast University, China. He is a product manager at ZTE Cloud Computing & IT Institute and is responsible for R&D and management of mobile internet and mass-data processing products. He has a deep understanding of mass-data processing, search engines, parallel calculation, and mobile terminal management.

DOI: 10.3939/j.issn.1673-5188.2014.01.009
http://www.cnki.net/kcms/detail/34.1294.TN.20131203.1814.002.html, published online December 3, 2013

A User-Recommendation Method Based on Social Media

Hong Chen[1], Shengmei Luo[1], Lei Hu[1], and Xiuwen Wang[2]
(1.ZTE Corporation, Nanjing 210012, China;
2. National Computer network Emergency Response technical Team, Beijing100029, China)

Abstract

User - analysis techniques are mainly used to recommend friends and information. This paper discusses the data characteristics of microblog users and describes a multidimensional user recommendation algorithm that takes into account microblog length, relativity between microblog and users, and familiarity between users. The experimental results show that this multidimensional algorithm is more accurate than a traditional recommendation algorithm.

Keywords

social media; user recommendation; information recommendation; relation analysis

1 Introduction

With the rapid development of the Internet, the manner of exchanging information has been changing. In the Web 1.0 era, information was propagated through simple static webpages; however, in the Web 2.0 era, information is published, propagated, and commented on dynamically through user relationship chains. Web 2.0 is like a visual society in which every user is able to freely share information, and the effects of interaction are getting more attention. Users are not only browsers but also producers of information. In the Web 1.0 era, websites were mainly news portals (e.g. Sohu.com, and Sina.com) and search engines (e.g. Google and Baidu). In the Web 2.0 era, websites are mainly online communities, online applications, social websites, blogs, and Wikis.

The most important difference between Web 1.0 and Web 2.0 is the central position of users and user relationships. As in the real world, every user on the Internet has their own interests, which can be observed by looking at their online behaviors, and some users even form common-interest groups.

This paper discusses certain characteristics of Sina Weibo microblog users in order to determine the interest relationships, and behaviors (e.g. publishing, forwarding, commenting, adding contacts) of these users. This can help them better establish their own microblog relationship networks.

2 Related Works

Analyzing Weibo users includes analyzing their relationships, subjects talked about, interests, and ego-networks. The results of such analysis can be used for making recommendations, advertising, or providing information. Much research has been done on social networking technologies both domestically and internationally.

In general, mainstream recommendation algorithms are based on collaborative filtering, content, and graph.

2.1 Collaborative Filtering

Recommendation algorithms based on collaborative filtering may be either user-based (UserCF) or item-based (ItemCF).

The main idea behind a user-based collaborative filtering algorithm is that things liked by the user's friends might also be liked by the user themself. First, the algorithm finds a selection of users who have similar interests to the target user. Then from this selection, it finds common items that can be recommended to the target user. The main point of step one is to calculate the similarity between the interests of any two users. The collaborative filtering algorithm is mainly interested in the similarity between user behaviors.

ItemCF algorithms have been widely applied in industrial sector. The recommendation algorithms of Amazon, Netflix, Hulu, and YouTube are all based on ItemCF. The main idea behind ItemCF is that a user tends to prefer similar items. Therefore, the algorithm makes recommendations to the user according to past favorites. However, ItemCF does not use the content of items to calculate this similarity. The algorithm mainly analyzes the user's behavior in order to calculate the similarity between items. The algorithm calculates the similarity between items and generates a recommendation list based on similarities and the user's historical behaviors.

UserCF recommendations are based on hotspots containing small groups of users with similar interests. ItemCF recommendations are based on the user's historical interests. UserCF recommendations are more social and reflect the popularity of items in a given small group of users. ItemCF recommendations are more personal and reflect a user's own historical interests.

2.2 Content-Based Recommendations

Content-based recommendation algorithms theoretically derive from information retrieval [1], [2] and information filtering [3]. They do not require a user to evaluate goods but rely on machine learning to extract the user's interests from descriptive content. The algorithm extracts the characteristics of items for which the user has given feedback and models the user's interests, calculates the similarity between the user's interests and items, and makes a recommendation.

Content-based recommendation algorithms make use of a user's historical data, and the user's interests may change in line with their preferences. The merits of this kind of algorithm are [4], [5]:

- There are no cold-start or sparse matrix problems.
- Recommendations can be made to users with special interests.
- New or unpopular items can be recommended.
- The contents of recommended items can be listed, and the results can be explained.
- Content-modeling technology is relatively mature.

The drawbacks of content-based recommendation algorithms are that they require content to be easily extracted into meaningful features; they require the content to be well-structured; and the user's interests must expressed in terms of the content's characteristics. Also, it cannot draw on the judgment of other users.

2.3 Latent-Factor Model

The latent-factor model is used in both recommendation systems and machine learning. It is an effective way of calculating the semantic distance for the purpose of classification and clustering. The main idea behind a latent-factor model is to link the user's interests to items through latent factors. Latent-factor technology has produced many well-known models and methods, including pLSA, LDA, latent class model, latent topic model, and matrix factorization. These are the same in essence, and some can be used in personalized recommendation systems.

3 User Recommendation

The main purpose of user recommendation is to recommend new friends to a target user according to the user's existing friends and historical behavior. This increases the overall density and activity of social networks.

In a social network, the user-recommendation algorithm is called link prediction. In [6], the relationship between the prediction methods of various users' friends was studied. Here, we introduce a user recommendation algorithm based on social network diagram and content matching.

The main idea behind a content-based matching algorithm is that friends' content is similar/related to their personal attributes (e.g. company, school, labels, location, IP).

User recommendation based on social network graphs has been widely used so that friends of the target user's friends may be recommended to the target user themself. Relationships on microblogs can be roughly categorized as following, followed, and mutual following. A suitable policy, therefore, is to recommend by followings or mutual followings. This method is used in Sina Weibo. However, recommendation by followings is flawed in that celebrities are always on the top. Recommendation by mutual followings is much better because the number of users who mutually follow a public celebrity is small.

This paper describes a user-recommendation algorithm that can be used online. We combine the two methods previously discussed in order to reduce computational complexity and make the algorithm more suitable for online use. For example, a user may be following 500 other users. In order to make u friend recommendations, it is very important to choose candidates.

If the pool of candidates is all the people on the microblog website, then this algorithm becomes too complicated [7], [8]. Using the conventional method, u's followers and followers of followers are candidates. In this way, time complexity is greatly reduced, but database operations still take a lot of time. In order to further reduce complexity, we propose a combined method shown in **Algorithm 1**.

Algorithm 1. A combined user-recommendation algorithm

1. Sort the followers of the followers of user u according to the number of followers. Assume that the sorted users are $re_{u1}, re_{u2}, \ldots, re_{uN}$, where N is the total number of candidates.
2. Divide the candidates into celebrities and non-celebrities according to the number of fans and whether the user has been certificated. If a user has been certificated and has more than 500,000 followers, they are marked as a celebrity. Recommend n_1 non-celebrities in descending order $re_{u1}, re_{u2}, \ldots, re_{uN}$ according to the number of indirect followings. Recommend n_2 celebrities in descending order $re_{u1}, re_{u2}, \ldots, re_{uN}$ according to the number of indirect followings and the celebrity's popularity.
3. Through these two steps, we can recommend some friends by relationships, but this is not enough. From the remaining users, select M users as the recommendation candidates, and $M << N$. Assume that the user is in a school called $school_u$ and in a company called $company_u$.
4. Extract the school names and company names of M candidates, and select candidates who are in the same company and school as user u because they are colleagues or alumni. Eventually, these four recommendations are presented to the user for them to choose.

4 Experiments

4.1 Data Description, Experimental Strategy and Evaluation Methods

We crawled the social networks and microblogs of 103 users of Sina Weibo, in order to determine the effectiveness of our proposed algorithm. The data collected from these networks and microblogs included 30,609 users, 807,374 relationships, and 2,503,458 microblogs. The data was acquired on May 1, 2013. This data is the both the training data set and evaluation data set.

First, we took a selection of each user's followers and used the recommendation algorithm to produce a recommendation list. Then, we compared this list with the selected followers.

We assume the seed node for the entire experimental data set is U, i.e. the set of experimental users is U, and we make recommendations from three sets of users: professional, ordinary, or all users.

We randomly remove half the users from the set of users $Followee(u)$ that u is following. These deleted users comprise the set $T(u)$, and the remaining set is the training set $R(u)$.

The lists $\sum_{u' \in R(u)} Followee(u)$ of followers of every user in $R(u)$ are the candidate lists. Before making recommendations, we delete users who are already in $Followee(u)$, i.e. $\sum_{u' \in R(u)} Followee(u) - R(u) - u$, and we express this set as $S(u)$.

Using the recommendation algorithm, users in $S(u)$ are sorted in descending order in order to get the recommendation list, also expressed as $S(u)$.

We take $S(u)$ as a test result and $T(u)$ as the standard answer. By comparing the difference between $S(u)$ and $T(u)$, we can evaluate the recommendation algorithm's performance. The first N-user subsets of $S(u)$ are denoted $S_1^n(u)$. We use two evaluation indexes: precision rate and recall rate. The recall rate of the recommendation is given by

$$Recall = \frac{\sum_{u \in U} \left| s_1^n(u) \cap T(u) \right|}{\sum_{u \in U} \left| T(u) \right|} \tag{1}$$

The precision rate of the recommendation is given by

$$Precision = \frac{\sum_{u \in U} \left| s_1^n(u) \cap T(u) \right|}{\sum_{u \in U} \left| s_1^n(u) \right|} \tag{2}$$

For a comprehensive evaluation Top-N recommendation precision and recall rate, various list lengths N are selected, the precision rate and recall rate are calculated, and a precision/recall curve is drawn. In this experiment, N is taken as 5, 10, 15, 20, 25, 30, 35, 40, 45, 50.

4.2 Results

The hypothesis of our experiment is that multidimensional recommendation is better than one-dimensional recommendation. Users in the popular or expert set have more than 500,000 fans and are certificated. Users in the second set are ordinary users, and there are no popular or expert users in this set. Users in the third set are a mixture of popular and ordinary users. We make recommend from the three sets and evaluate these recommendations using evaluation indexes.

4.2.1 Popular or Expert Users

We select popular or expert users who were removed from the list of followees $T(u)$, also from $S(u)$, and selected Top-N qualified users to recommend as a result recommendation list. Then, we calculated the precision and recall rate using the evaluation formula. The results are shown in **Table 1**.

From Table 1, the precision and recall rates are 0.163 and 0.253, respectively, when N = 50. The precision and recall curves are shown in **Fig. 1**.

4.2.2 Ordinary Users

We select qualified ordinary users who were removed from $T(u)$ and $S(u)$. Top-N qualified users were selected for inclusion in a result recommendation list. Then, the precision and recall rate were calculated using the evaluation formula. The results are shown in **Table 2**.

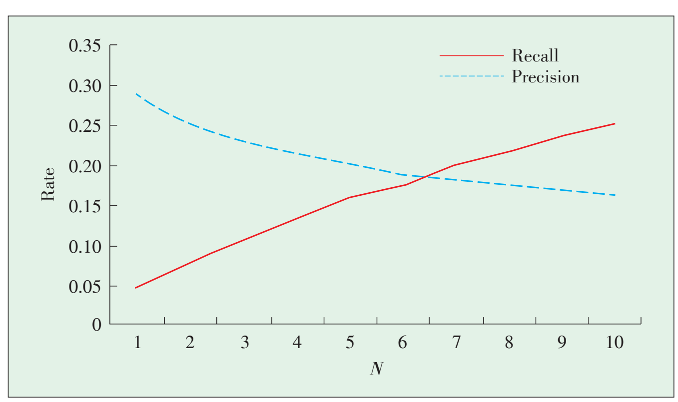
From Table 2, the precision and recall rates are 0.359 and 0.072, respectively, when N = 50. The precision and recall curves are shown in **Fig. 2**.

4.2.3 All Users

Here, recommendation results are not divided into different sets. We mix popular users with ordinary users and calculate

▼**Table 1. Recall and precision rates of popular users**

N	Recall rate	Precision rate
5	0.045	0.289
10	0.078	0.251
15	0.107	0.229
20	0.133	0.214
25	0.159	0.203
30	0.175	0.188
35	0.201	0.185
40	0.219	0.178
45	0.238	0.170
50	0.253	0.163



▲**Figure 1. Precision and recall curves for popular or expert users.**

the precision and recall rates using the evaluation formula. The results are shown in **Table 3**.

From Table 3, the precision and recall rates are 0.329 and 0.058, respectively, when $N = 50$. The precision and recall curves are shown in **Fig. 3**.

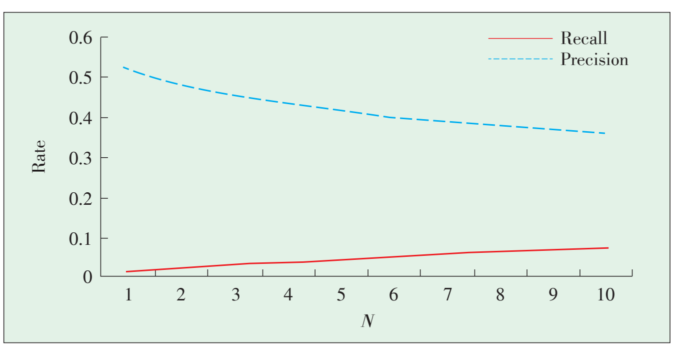
4.2.4 Comparison

The precision and recall rates of the three Top-N recommendation strategies are shown in **Fig. 4**.

From Fig. 4, the precision rate when recommending an ordinary user is better than that when recommending a popular user or any user. The precision rate when recommending popular users is lower than that when recommending ordinary users. The main reason for this is that the proportion of popular users on microblog websites is very small. The total number of users is about 500 million, but the number of popular users is less than 3 million. Also, people are usually already aware of popular users before using a microblog websites. Therefore, they may search out and follow popular users through other channels, such as a search engine. A specific user is also likely to follow a limited number of stars or experts and not easily follow

▼**Table 2. Precision and recall rates for ordinary users**

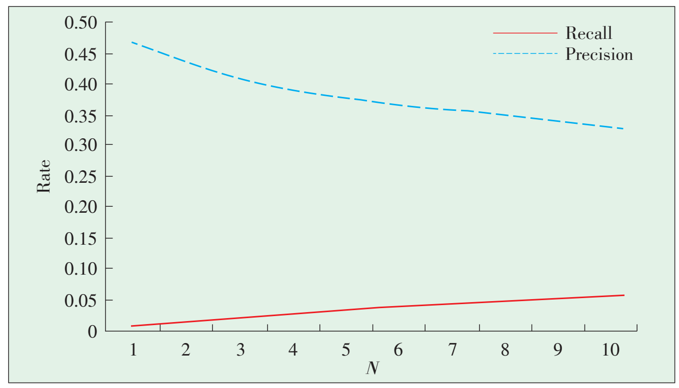
N	Recall rate	Precision rate
5	0.010	0.520
10	0.019	0.483
15	0.028	0.458
20	0.035	0.435
25	0.042	0.417
30	0.048	0.401
35	0.055	0.388
40	0.061	0.380
45	0.067	0.368
50	0.072	0.359



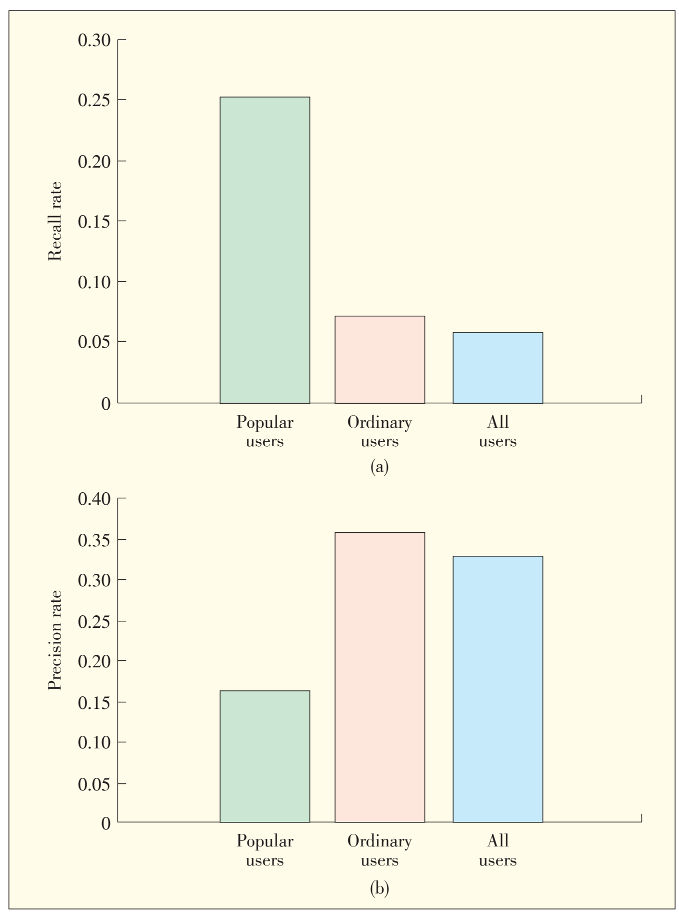
▲**Figure 2. Precision and recall curves for ordinary users.**

▼**Table 3. Precision and recall rates for all users**

N	Recall rate	Precision rate
5	0.008	0.464
10	0.015	0.437
15	0.022	0.410
20	0.028	0.388
25	0.034	0.378
30	0.039	0.364
35	0.044	0.357
40	0.049	0.348
45	0.054	0.338
50	0.058	0.329

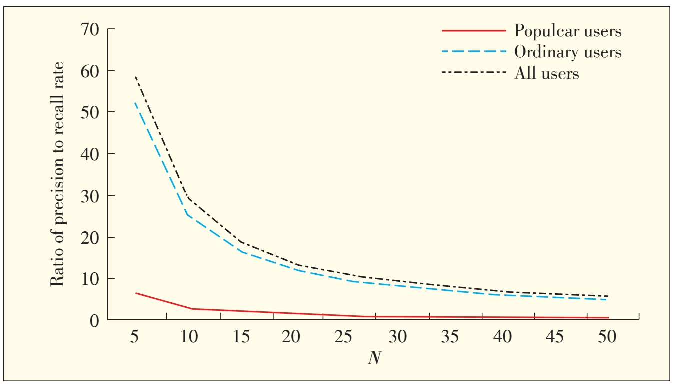


▲**Figure 3. Precision and recall curves for all users.**



▲**Figure 4. (a) Recall rates and (b) precision rates for the three Top-N recommendation strategies.**

other popular users. Therefore, the precision rate when recommending popular users is relatively low. The main reason the precision rate when recommending any user is lower than when recommending an ordinary user is that popular users are included in the mix or all users. In Fig. 4a and b, the higher the precision rate, the lower the recall rate. This is also clearly shown in the graph of the ratio of precision to recall for the three recommendation strategies (**Fig. 5**).



▲**Figure 5. Ratio of precision to recall for the three recommendation strategies.**

The advantages of multidimensional recommendation are that it avoids deviation caused by popular users and is more compatible with user psychology.

5 Conclusion

Hundreds of millions of people use social media, represented in this paper by Sina Weibo. As users publish, communicate and review, they generating a lot of content and the relationship between them expands. All of social media is connected by user entities whose offline behaviors gradually migrate online.

The algorithm in section 3 is effective for recommending new friends to a target user. The time and space complexity of the algorithm meets the needs of online applications. However, further work needs to be done on increasing the algorithm's accuracy.

References

[1] F. Diaz, "Regularizing query-based retrieval scores," *Information Retrieval*, vol. 10, no. 6, pp. 531-562, 2007. doi: 10.1007/s10791-007-9034-8.

[2] S. Clemencon, G. Lugosi, and N. Vayatis, "Ranking and empirical minimization of U-statistics," *The Annals of Statistic*, vol. 36, no. 2, pp. 844-874, 2008. doi: 10.1214/009052607000000910.

[3] N. Belkin and B. Croft, "Information filtering and information retrieval," *Comm. ACM*, vol. 35, no. 12, pp. 29-37, 1992. doi: 10.1145/138859.138861.

[4] B. B. Ana, C. M. Enrique, C. B. Juan, R.L. Marta, A. M. F. Fernando and P. Ana, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290-4311, 2010.

[5] M. Pazzani, "A framework for collaborative, content-based, and demographic filtering," *Artifical Intelligence Rev.*, vol. 13, no. 5-6, pp. 393-408, 1999. doi: 10.1023/A:1006544522159.

[6] *The Link Prediction Problem for Social Networks* [Online]. Available: http://www.cs.cornell.edu/home/kleinber/link-pred.pdf

[7] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock, "Methods and metrics for cold- start recommendations," in *Proc. 25th Ann. Int'l ACM SIGIR Conf.*, 2002, pp. 253-260. doi: 10.1145/564376.564421.

[8] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, no. 1, pp. 103-145, 2005. doi: 10.1145/1055709.1055714.

Manuscript Received: August 20, 2013

Biographies

Hong Chen (chen.hong3@zte.com.cn) received her BS degree from the Department of Information Engineering, Nanjing University of Posts and Telecommunications, in 2007. She currently works as a senior research engineer at ZTE Corporation. Her research interests incloud social network analysis and Intelligent Question Answering. She holds four patents .

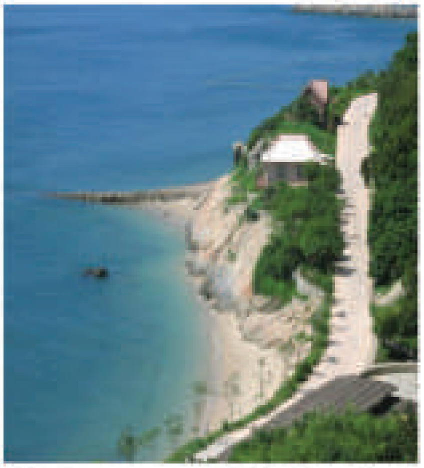
Shengmei Luo (luo.shengmei@zte.com.cn) received his MS degree in telecommunication and electronics engineering from Harbin Institue of Technology, China, in 1996. He is a chief architect at ZTE Corporation. His research interests include cloud mobile Internet, cloud computing, and industrial application of planning. He has published 20 papers.

Lei Hu (hu.lei2@zte.com.cn) received his MS degree from the Laboratory of Intelligent Recognition and Image Processing, Beihang University, in 2008. He is a senior research engineer in the area of mass data analysis at ZTE Corporation. His research interests include data mining, information retrieval, and social network analysis. He holds three patents and has published four papers in these fields.

Xiuwen Wang (wxw@cert.org.cn) received her PhD degree in computer system architecture from the Institute of Computing Technology, Chinese Academy of Sciences, in 2010. She now works as a senior research engineer in the China National Computer Network Emergency Response Team (CNCERT). Her research interests include cloud social network analysis and data analysis.

Call for Papers

Maoming, China
14-16 August, 2014
http://chinacom.org/show/home



HIGHLIGHTS:

- Chinacom 2014 will ***feature keynote speeches*** given by:

Prof. Vincenzo Piuri, IEEE Fellow, Università degli Studi di Milano, Italy
Prof. Nei Kato, IEEE Fellow, Tohoku University, Japan
Prof. Hanna Bogucka, Poznan University of Technology, Poland

- Chinacom 2014 will offer several workshops (number to be finalized); see website for further information and calls for workshop papers.
- All accepted and presented papers will be included in the ChinaCom 2014 Conference Proceedings, and will be submitted for inclusion in ***IEEE Xplore and EI*** Compendex.
- Top papers will be recommended for publication in the ***special issues of MONET and SCN (SCI indexed)*** after further extension as well as EAI Endorsed Transactions on Industrial Networks and Intelligent Systems.
- The Organizing Committee will select the best papers of each symposium and present the authors with ***best paper awards*** during the conference.
- The conference will be held in ***Maoming***, the oil city of the south with beautiful landscape in mainland China, from 14-16, August, 2014.

Scope of the Conference

ChinaCom is endorsed by the European Alliance for Innovation (http://eai.eu/), a leading community-based organization devoted to the advancement of innovation in the field of ICT. The aim of ChinaCom is to bring together Chinese and international researchers and practitioners in networking and communications under one roof, building a showcase of these fields in China. ChinaCom is being positioned as the premier international annual conference for the presentation of original and fundamental research advances in the field of communications, networks, and internet applications. The conference will consist of technical symposia, panels, tutorials, and workshops. All submissions to this conference will be subject to rigorous peer review before included in the technical sessions.

Steering Committees:
Imrich Chlamtac, CREATE-NET
Ya-Bin Ye, Huawei Europe Research Center
Zheng Zhou, Beijing University of Posts and Telecommunications
Bo Li, Hong Kong University of Science and Technology
Andreas F. Molisch, University of South California
Jun Zheng, Southeast University
Zhi-Feng Zhao, Zhejiang University
Hsiao-Hwa Chen, National Cheng Kung University

General Chair:
Qinghua Zhang, Guangdong Univ. of Petrochemical Technology, China

General Co-Chair:
Vincenzo Piuri, Università degli Studi di Milano, Italy
Lei Shu, Guangdong Univ. of Petrochemical Technology, China

TPC Chair/TPC Co-Chairs:
Yan Zhang, Simula Research Laboratory, Norway
Danny Tsang, Hongkong University of Science and Technology
Chonggang Wang, InterDigital Communications, USA

Publication Chair(s):
Weiqiang Xu, Zhejiang Sci-Tech University, China

Publicity Chair(s):
Trung Q. Duong, Queen's University Belfast, UK
Chunsheng Zhu, University of British Columbia, Canada
Kun Wang, Nanjing Univ. of Posts and Telecommunications, China
Tomohiko Taniguchi, Fujitsu Labs. LTD., Japan

Workshops Chair(s):
Noel Crespi, Institute Mines-Telecom, Telecom SudParis, France
Panlong Yang, PLA University of Science and Technology, Nanjing
Xiaohu Ge, Huazhong University of Science and Technology, China

Demos and Tutorials Chair(s):
Rongxing Lu, Nanyang Technical University, Singapore
Guangjie Han, Hohai University, China

Local Chair:
Mei Liu, Guangdong Univ. of Petrochemical Technology, China
Zhongming Yang, Guangdong Univ. of Petrochemical Tech., China

Web Chair:
Haiming Zhang, Chinese Academy of Sciences, China
Aisong Qin, Guangdong Univ. of Petrochemical Technology, China

Conference Coordinator:
Ruzanna Naiarvan. EAI

Publications

All accepted and presented papers will be included in the ChinaCom 2014 Conference Proceedings, and will be submitted for inclusion in ***IEEE Xplore and EI*** Compendex. Papers accepted and presented in ChinaCom2006-2013 have been indexed by EI entirely. Selected best and outstanding papers will be considered for publication in special issues of ***SCI-indexed Journals MONET and SCN*** after further extension as well as EAI Endorsed Transactions on Industrial Networks and Intelligent Systems.

Important Dates

Paper submission: 30 April, 2014
Author notification: 20 June, 2014
Camera-ready:1 July, 2014
Conference: 14-16 August, 2014

Submissions

Please refer to http://chinacom.org/show/initial-submission for further information on submission format and requirements.

ZTE Communications Guidelines for Authors

• Remit of Journal

ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro−optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

• Manuscript Preparation

Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 4000 to 7000, and no more than 6 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

• Abstract and Keywords

Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self−contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Five carefully chosen keywords must be provided with the abstract.

• References

Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially in−text and listed in corresponding order at the end of the paper. References that are not cited in−text should not be included in the reference list. References must be complete and formatted according to IEEE Editorial Style www.ieee.org/documents/stylemanual.pdf. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

• Copyright and Declaration

Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co−authors, b) the manuscript has not been published elsewhere in its submitted form, c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co−authors should not exceed five.

• Content and Structure

ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

• Peer Review and Editing

All manuscripts will be subject to a two−stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

• Biographical Information

All authors are requested to provide a brief biography (approx. 150 words) that includes email address, educational background, career experience, research interests, awards, and publications.

• Acknowledgements and Funding

A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

• Address for Submission

magazine@zte.com.cn
12F Kaixuan Building, 329 Jinzhai Rd, Hefei 230061, P. R. China