Enabling Vehicular Networking in the MobilityFirst Future Internet Architecture[⋆]

Akash Baid[*], Shreyasee Mukherjee[*], Tam Vu[*], Sandeep Mudigonda[†],
Kiran Nagaraja[*], Junichiro Fukuyama[‡], Dipankar Raychaudhuri[*]

[*]WINLAB, Rutgers University, [†]Dept. of Civil Engg., Rutgers University [‡]Toyota InfoTechnology Center USA

***Abstract*—Vehicular networking, both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), is an increasingly important usage scenario for future mobile Internet services. Radio technologies such as 3G/4G and WAVE/802.11p now enable vehicles to communicate with each other and connect to the Internet, but there is still the lack of a unifying network protocol architecture for delivery of services across both V2V and V2I modes. The MobilityFirst future Internet architecture, discussed in this paper, is a clean-slate protocol design in which the requirements of untethered nodes and dynamically formed networks are considered from the ground-up, making it particularly suitable for vehicular applications. Here we describe the vehicular networking specific features and protocol design details of the architecture and present evaluation results on performance and scalability.**

I. Introduction

Vehicular networking represents an extreme point in the mobile network design space because of fast node mobility involving variable speed, intermittent connectivity, and high uncertainty in network load due to variable node density and data traffic demands [1], [2]. The IEEE 1609 ITS standard (WAVE) [3] offers an IP-less messaging alternative (alongside a TCP/IP stack), and addresses several PHY/MAC layer challenges for time-sensitive applications. The dual stack approach, however, presents a complex service interface for applications which then require awareness of both networks and to actively switch between them to achieve service objectives. Our proposed Internet-wide architecture, MobilityFirst [4], presents a unifying network architecture, protocol stack and service API that migrates smoothly from fully connected to weakly connected to ad hoc network environments associated with vehicular systems [5].

In this work, we describe MobilityFirst's support for four key communication patterns that we believe together cover most vehicular applications' needs: (i) interactive communication between infrastructure and vehicle, (ii) data dissemination from infrastructure to vehicles (I2V), (iii) vehicle-to-infrastructure (V2I) sensor data upload, and (iv) vehicle-to-vehicle (V2V) messaging. Due to space limitations, we only describe and present evaluation results for the first communication class in this paper. We have performed large scale and realistic simulations of vehicular mobility on both highway (New Jersey Turnpike) and urban scenario (Jersey City, NJ) to show feasibility for mobility handling at-scale. We also present results from detailed NS3 simulations of file downloading and web-browsing applications over MobilityFirst that show significant performance gains (in terms of throughput) and improvements to user experience (in terms of browsing delays) over versions that ran on TCP/IP.

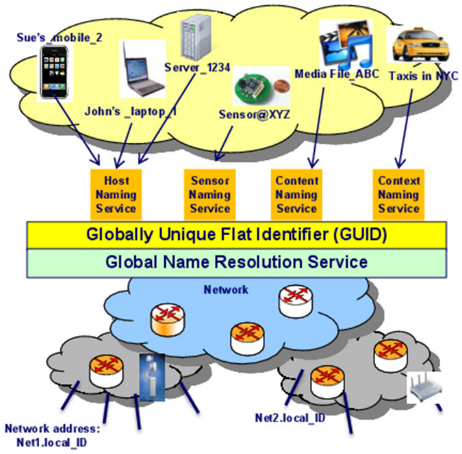


Fig. 1: Separation of identification and network location in the MobilityFirst architecture

II. Vehicular Networking through MobilityFirst

A. Overview of MobilityFirst

The MobilityFirst architecture is built upon a new name-based service layer which serves as the "narrow-waist" of the protocol stack. The name-based service layer uses flat globally unique identifiers (GUIDs) for all network attached objects - from a simple device such as a smartphone, a person, a vehicle, a group of vehicles, a piece of content, and even context, as shown in Fig. 1. For more details on the design goals, key architectural concepts, protocol details, and prototyping efforts, please refer to our earlier works [4].

A GUID can be assigned to a network object by one of multiple name certification services (NCSs), and is derived through a cryptographic hash of the public key that corresponds to that object. The GUID being directly derived from the public key gives it a self-certifying property, i.e. authenticating a node does not require an external authority [6]. This feature solves an important problem in vehicular networks

[⋆]Research supported by NSF Future Internet Architecture (FIA) grant CNS-1040735

978-1-4673-5828-6/13/$31.00 ©2013 IEEE

where communication to a third-party server is often not possible or introduces substantial delay to critical applications.

The dynamic mapping of GUIDs to NAs is made possible through a logically centralized, but physically distributed infrastructure, which we call the global name resolution service (GNRS). Details about the design, prototype, and evaluation of the GNRS can be found in [7]. A key feature, and one which is crucial to supporting the wide variety of vehicular applications, is the use of service flags in the API (and service identifier (SID) in the packet). With support for different kinds of services, network entities can tailor the behaviour of the forwarding and compute (i.e., those which process the packet's payload) elements to suit the applications that they support.

B. I2V Interactive Applications through MobilityFirst

As a specific example of the I2V applications, let us consider a vehicle equipped with both WiFi and 4G cellular radios driving down a highway while streaming a video to be viewed in the in-car entertainment system. Fig. 2 illustrates this scenario and also shows the connectivity pattern. For requesting the video stream, the in-car application issues a fetch request: $get(GUID_c, ...)$. The downlink response from the nearest server: $send(GUID_v, message, ...)$, is first resolved to get the current network attachment point of the vehicle (NA_1), and then routed to that network, keeping both $GUID_v$ and NA_1 in the header.

In our example, when the vehicle disconnects from NA_1, the downlink packets are not discarded but stored locally. When the vehicle reconnects to the Internet through NA_2, via WiFi, an update is sent to the GNRS to reflect the new attachment point. Meanwhile, NA_1 queries the GNRS for any updates on $GUID_v$, and having learnt that the vehicle is now in NA_2, sends out the packets to that network. New packets from the server are automatically sent to NA_2 owing to the GNRS lookup before transmission. The GNRS also enables seamless multihoming support - going ahead with our example, when the vehicle is in a region of both WiFi and cellular connectivity, it updates the GNRS with two network attachment points: $<GUID_v:NA_2,NA_3>$. The owner of the GUID can also specify their policies, such as 'send to both', 'send to any', etc. when requesting packets from a source. Thus the downlink packets, in this region, can be sent over both the interfaces for increased capacity or increased resilience. Finally, when the vehicle exits the WiFi region, the GNRS state is updated again to reflect connectivity to only NA_3, and remaining packets are sent through this network.

III. Evaluation Results

A. Scalability of GNRS Updates

In order to manage mobility at the network layer, each node (or a designated proxy) must update the physically distributed GNRS servers whenever it changes its point-of-attachment to the Internet. This raises a scalability concern in the vehicular environment - since vehicular nodes switch basestations/access points very frequently, would the resulting GNRS update traffic create significant overhead?

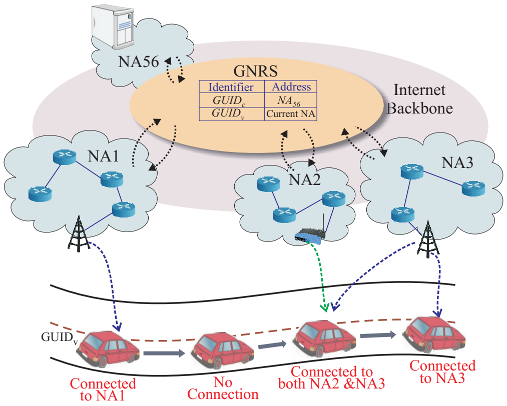


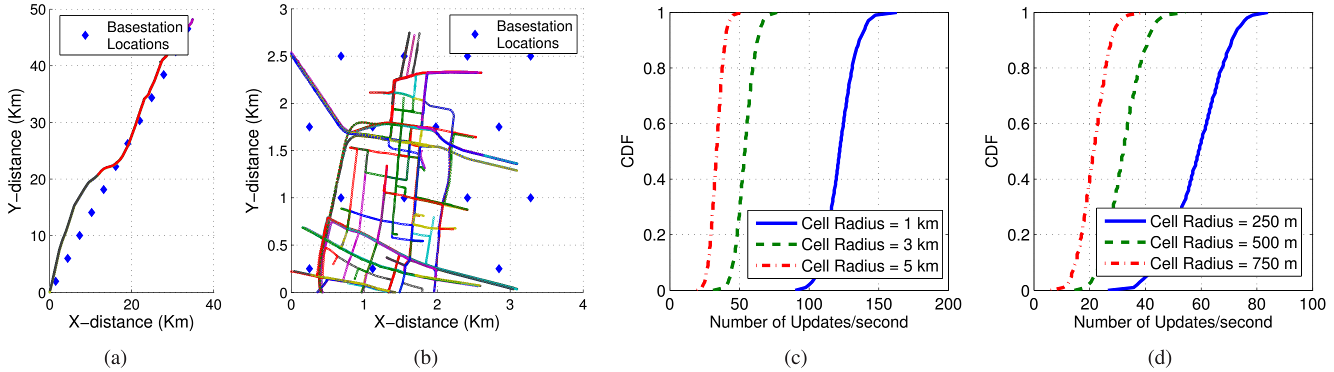
Fig. 2: Disconnection tolerance and in-network mobility management through MobilityFirst

In this section we address this issue through an analysis of the number of updates generated in two real road segments - (i) a 25 mile stretch of the New Jersey Turnpike, referred as the NJTPK trace; (ii) a 3 square mile urban area of Jersey City in New Jersey, referred as JC trace. To realistically analyze vehicular mobility, we use a well-calibrated model of the New Jersey Turnpike and Jersey City that is built in a powerful microscopic simulation tool, PARAlell Microscopic Simulation (PARAMICS) [8]. Further details about the simulation platform are available from our earlier works [8], [9]. We assume three different values for the cell radius in the NJTPK trace: 1, 3, and 5 km, and assume base stations are placed along the highway. Urban areas, in contrast, are more densely covered, often with low powered small cells. Thus we assume regular hexagon cell deployment with cell radius: 250, 500, and 750 meters in the JC trace. Example traces of 100 randomly selected vehicles are shown in Figs. 3(a) and 3(b).

Figs. 3(c) and 3(d) show the cumulative distribution function of the number of updates generated per second in the NJTPK and the JC vehicular traces. The plots show that even for a dense vehicular network and worst case assumptions about the frequency of updates, a maximum of around 160 and 80 updates occur per second in the two traces respectively. Since each update would typically correspond to about 40-100 bytes of transmission (the exact size depends on the nature of the network address space in use), this would lead to a traffic overhead of less than 16 KBps.

B. Throughput at Vehicular Speeds

Here we compare the performance of the MobilityFirst architecture with the current TCP/IP based Internet access for vehicular nodes, through a detailed NS3 simulation. The simulation topology consists of a single mobile client with 802.11 radio moving along a straight road, with access points deployed along the road at random inter-AP distances d (picked from a uniform random distribution between 300-



(a)

(b)

(c)

(d)

Fig. 3: PARAMICS Results:(a) and (b) Movement trace of 100 randomly selected vehicles along with an instance of basestation placement (cell radius: 3 km and 500 m), (c) and (d) CDF of the number of updates/sec across all basestations, from NJTPK and JC traces respectively

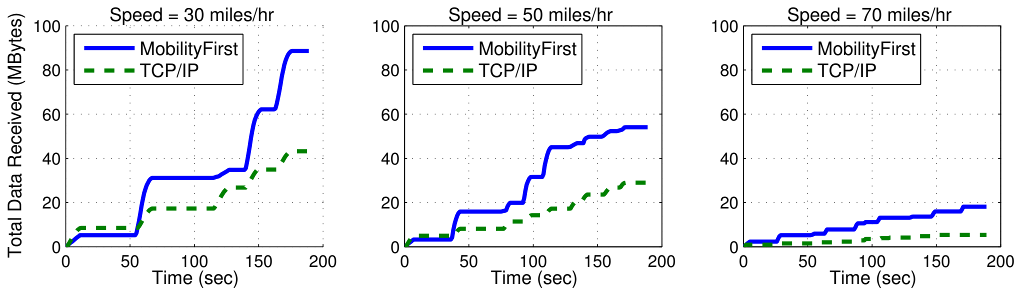


Fig. 4: Aggregate throughput over time for MobilityFirst and TCP/IP across different vehicular speeds

500m). A remote data server is assumed to be connected to the access points through the back-end wired network. The client requests a large file from the server at the start time of the simulation, and transmission of the file continues till the end.

Fig. 4 shows the total data received by the client as a function of the simulation run-time, for three different vehicular speeds. Since the dwell time of the moving node inside the range of APs reduces with increasing speed of the node, the overall throughput reduces for both MobilityFirst and TCP/IP. However, after each period of no connectivity (identified by the flat horizontal portions of the curves), we can observe a jump in the MobilityFirst curve across all speeds. This gain can be traced to two key differences between MobilityFirst and TCP/IP. First, when the node disconnects from an AP, the packets destined for it are stored locally at the last AP instead of being dropped, and are sent quickly to the next AP when the node connects there. And second, the amount of 'useful time' spent in each AP is increased since the node retains its GUID as it traverses multiple APs.

IV. Conclusion

In this paper, we propose the use of the MobilityFirst future Internet architecture as a unified solution for supporting different types of vehicular networking applications. We provide a brief outline of the mobility-centric viewpoint of the MobilityFirst project, along with the details on how vehicular applications can be efficiently supported through the proposed architecture. In particular, disconnection-tolerance, storage-capable routing, in-network mobility management, multi-homing support, compute layer functionality, and disconnected mode operations are highlighted. We present results from two different simulation studies: (i) through realistic large-scale vehicular traces, we show the scalability of the proposed GNRS; and (ii) through a detailed NS3 simulation, we quantify the advantages of MobilityFirst over existing TCP/IP mechanisms.

References

[1] M. C. Chan and R. Ramjee, "TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation," *Wireless Networks*, pp. 81–97, 2005.

[2] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay- and Disruption- Tolerant Networking," *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, 2006.

[3] "IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments," Available from IEEE Standards.

[4] MobilityFirst Future Internet Architecture Project, http://mobilityfirst.winlab.rutgers.edu/.

[5] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR: Generalized storage-aware routing for MobilityFirst in the future mobile Internet," in *Proceedings of MobiArch'11*, 2011, pp. 19–24.

[6] D. G. Andersen et al., "Accountable Internet Protocol (AIP)," in *Proc. ACM SIGCOMM'08*, Aug. 2008.

[7] T. Vu et al., "DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet," in *Proceedings of ICDCS '12*, 2012, pp. 698–707.

[8] K. Ozbay and S. Mudigonda, "Simulated Vehicle Trajectory Data," Rutgers Intelligent Transportation Systems (RITS) Laboratory, Rutgers University, Piscataway New Jersey, Dec. 2012.

[9] K. Ozbay, S. Mudigonda, and B. Bartin, "Microscopic Simulation and Calibration of an Integrated Freeway and Toll Plaza Model," Presented at the 85th TRB Annual Meeting, 2006, Washington, D.C.