Cloud-based Handoff as a Service for Heterogeneous Vehicular Networks with OpenFlow

Ke Xu[1], Ryan Izard[1], Fan Yang[1], Kuang-Ching Wang[1], and Jim Martin[2]

[1]Department of Electrical and Computer Engineering
Clemson University
Clemson, SC, USA
{kxu, rizard, fany, kwang}@clemson.edu

[2]School of Computing
Clemson University
Clemson, SC, USA
jim.martin@cs.clemson.edu

***Abstract*— This paper proposes a cloud-based architecture for enhancing the performance and capacity of vehicular networks of potentially multiple different wireless technologies. The approach addresses the well-known limitations of today's vehicle-initiated as well as base station-assisted handoff solutions; the former is reactive, therefore slow and inefficient, while the latter is mostly limited to within networks of a single technology. The handoff-as-a-service (HaaS) architecture leverages a cloud system's abundant computing and data storage resources to establish a database of key network properties and configuration options. By abstracting different networks' characteristics into a common set of descriptors, the database can aggregate and share properties of networks of different technologies. Leveraging network awareness of a wider scope, the HaaS service can further analyze optimal network configurations considering global efficiency and individual client requirements. The HaaS service in the computing cloud computes optimal handoff strategies on behalf of the vehicles, and OpenFlow is used to control both the vehicle and infrastructure side network interfaces seamlessly across multiple interfaces of different wireless technologies. This paper presents the proposed system architecture, its key components, and how they can be experimentally studied over National Science Foundation's Global Environment for Network Innovations (GENI) testbed. Experiment results on PC Engine device show the feasibility and advantage of the proposed handoff solution.**

Keywords*— *handoff, database, cloud, vehicular networks, heterogeneous networks, GENI

I. INTRODUCTION

A myriad of wireless communication technologies has brought broadband Internet access to automobiles on the road. Maintaining reliable and high bandwidth connectivity is a difficult problem due to the rapidly varying wireless communication channel conditions between moving vehicles and the infrastructure. At the same time, if one considers the aggregate bandwidth and coverage of all diverse types of wireless networks, they actually hold the potential of achieving reliable and high bandwidth network connectivity in most urban and suburban areas. Recent studies have attempted to make joint use of multiple technologies to improve the continuity and quality of vehicles' network connectivity, e.g., [1,2]. The former adopts a reactive approach to always use a second radio to scan for the next available base stations, while the latter adopts proactive data delivery on both Wi-Fi and WiMAX networks to maintain continuous data delivery. A reactive approach by nature would require sufficient scan time for discovery. A proactive approach, on the other hand, relies on knowledge of the availability and quality of network options, which are the subject of infrastructure-assisted vertical handoff solutions. To date, there have not been standard methods for cross-technology handoff and simultaneous usage of heterogeneous wireless interfaces. Addressing satisfactory handoff with low latency and minimal throughput disturbance is essential in achieving end-use quality of service (QoS) satisfaction and network resource utilization efficiency.

Traditional horizontal handoff mechanisms (among base stations of the same technology) are mostly based on received signal strength (RSS), e.g. [3]. In [4], a pre-scanning method is proposed to reduce the number of handoffs and maintain load balance among access points (APs) considering highly mobile vehicular scenarios. On the other hand, strategies for vertical handoff between different access networks are mostly based on performance, availability or economic factors. RSS-based algorithms [5,6] and cost-based algorithms [7] are two major classifications for making handoff decisions.

Two important challenges still exist for handoff in vehicular networks. First, the difficulty presented to avoid frequent unnecessary handoffs caused by vehicular environments and mobility. The ping-pong effects, yielding unnecessary handoffs back and forth between different road-side units (RSUs[1]), usually are caused by three factors: a) fluctuating instantaneous RSS values, mainly resulting from fast fading which is demonstrated to influence channel characteristics extensively for vehicular environments [8]. b) vehicle moving around overlapping coverage areas between two adjacent networks. c) vertical handoff from a wide-coverage network to a newly discovered local-coverage network only followed by another vertical handoff back to the original network quickly. The ping-pong effects obviously impose heavy handoff process loads and signaling overhead to the network. The second challenge for handoffs is the difficulty in the design of optimal handoff strategies considering both global network efficiency and individual client requirements. The vehicle-initiated handoff, e.g. [7], and infrastructure-assisted handoff, e.g. [4], are two typical types of handoff solutions in vehicular networks. As a starting point of satisfying an individual vehicle's demands as much as possible, e.g. [7], the distributed vehicle-initiated handoff may result in inefficiency of allocating network resources due to each one self-selecting its own optimal RSU selfishly. While the

[1]The paper uses the term "RSU" to represent all types of statically deployed infrastructures in heterogeneous networks, including access points in wireless local area networks and base stations in cellular networks.

considerations of global network information, such as network load distribution in infrastructure-assisted handoff solutions [4], is usually limited to a single type of access network, due to the ease of cooperation and scheduling among homogeneous RSUs.

To address the handoff challenges described above in vehicular networks, this paper proposes a handoff-as-a-service (HaaS) architecture, which leverages cloud system's abundant computing and data storage to establish a database of key network properties and configuration options. By abstracting different networks' characteristics into a common set of descriptors, the database can aggregate and share properties of networks of different technologies. The HaaS service in the computing cloud computes optimal handoff strategies on behalf of the vehicles. OpenFlow is the underlying technology used to control the vehicle and infrastructure side network interfaces seamlessly across multiple interfaces of different wireless technologies.

The remainder of this paper is structured as follows: Section II explains the proposed system overview and the proposed HaaS architecture. Section III introduces future experimentation plans and on-board handoff performance using the Global Environment for Network Innovations (GENI) testbed. Finally, the paper is summarized in Section IV.

II. The Proposed Handoff-as-a-Service

This section firstly introduces the overview of the proposed cloud system, and secondly explains the HaaS service architecture by key components and functions.

A. System Overview

The proposed system is shown in Fig. 1. The overall system consists of a database server, multiple local RSUs, and active vehicles in the coverage. The database server is a key component in the proposed system, which takes charge of establishing, maintaining the database, computing optimal handoff strategies, and communicating with RSUs and vehicles. The server can be accessed by all the local RSUs and connects to Internet. Database updates are provided by both local resources including RSUs' and vehicles' information, and remote resources such as the Transportation Coordinators'-Automated Information for Movements System II (TC-AIMS II) [14] providing the road traffic flow information in the region. The database server has full knowledge of the network configurations and status and is able to compute handoff cost function and provide optimal handoff strategies for vehicles. All RSUs send updates of their status such as the number of active vehicles and traffic loads to the server, and periodically or intentionally broadcast beacons scheduled by the server, and push the handoff decisions to desired vehicles. The vehicles are equipped with GPS devices. The on-board device connecting to the network is assumed to be able to import data from the GPS and integrates different wireless access technologies. The vehicle periodically reports its status, including location, speed, RSS, and battery status to the associated RSU -- about 1~5 Hz based on GPS update rate. The updates will not add much traffic to the network since the packet typically is very short (less than 100 B) [10]. Eventually, the vehicle's device

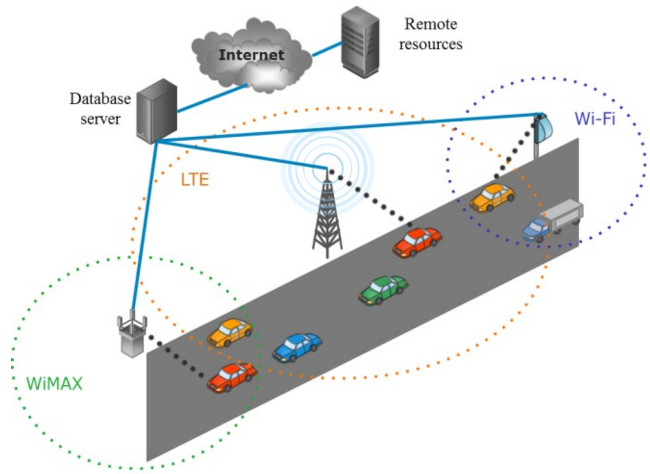

Figure 1. System overview

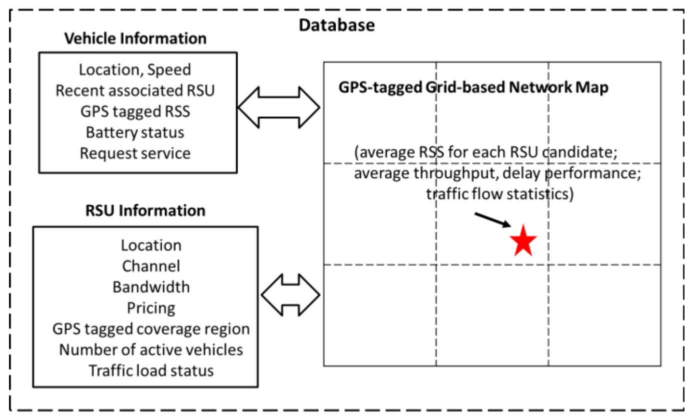

Figure 2. Database

executes handoff based on receiving server's handoff decision. Note here, if the vehicle cannot reach the server in case, the on-board device will make the handoff decision by itself based on signal strength as the regular way.

B. HaaS Architechture

This subsection first introduces what information is stored in the database; second, explains how to build and maintain the database; and finally, proposes the handoff decision algorithm using the database.

1) Information in the database

The server database consists of three parts: vehicle information, RSU information, and a GPS-tagged, grid-based network map, as illustrated in Fig. 2. The vehicle and RSU information is based on periodical updates from vehicles and RSUs, respectively. This information reflects the updated and changing status and conditions on individual vehicles and RSUs. The information will include time stamps. The grid-based network map records historical information, such as RSS, throughput, and traffic flow for each grid corresponding to GPS locations. The statistics are important for making handoff decisions. For example, using average normalized RSS based on most recent information in a grid for a certain RSU could indicate the average signal quality level for this RSU, thus avoiding instantaneous fluctuation. There might be situations however where older information can help to improve the accuracy of decision. For example, at 5:00pm each day at a particular location the RSS of an RSU drops by 20 dB. The network map and vehicle/RSU information sets are able to exchange information with each other.

2) Database maintenance

The database maintenance typically is divided into two steps: *preliminary collection* and *online update*. Preliminary collection is used for initially establishing the database, in particular for the network map. During the collection procedure, the testing vehicle records the RSS corresponding to particular RSUs mapped into grids, and then uses software tools to obtain performance samples. The database requests the traffic flow statistics from remote resources such as TC-AIMS II periodically and records the network map. Online update is for real-time updates of vehicle/RSU information in the database, which obtains the information from periodic update messages sent by active vehicles on the road and RSUs. Some updating information, e.g. RSS, number of active vehicles, is added to historical records in the network map.

3) Handoff decision algorithm

The handoff decision is formulated as a fuzzy logic problem. Fuzzy logic is used to represent the imprecise information of some attributes and user preferences. Each attribute is ranked into several ordered levels by being normalized or graded. The decision metrics are separated into three major classes: availability, performance, and pricing, which are denoted by the notation M_{1i}, M_{2i}, M_{3i} for any candidate RSU i. Availability metric considers attributes <average RSS, battery status, sojourn time>; performance metric considers <average throughput, average delay, traffic flow, number of active vehicles in the cell>; pricing metric considers <per-unit cost, service type>. For each metric class, the score for a candidate RSU i is determined by the weighted sum of all the attribute values, given as

$$M_{ji} = \sum_{k=1}^{n} w_k r_k \quad (1)$$

where r_k is kth attribute of metric class, w_k is the weight assigned for kth attribute, n is the number of attributes, and j=1,2,3 is the index of metric class. Then by calculating the weighted sum of all three metric classes and selecting the maximum the best candidate network is

$$I^* = \operatorname{argmax}_{i \in N} \sum_{j=1}^{3} w'_j M_{ji} \quad (2)$$

where w'_j is the weight assigned for jth metric class, and N is the number of RSU candidates.

III. Experiments using GENI Testbed

This section introduces the experimentation plans using GENI wireless testbed. Following this, database establishment and on-board devices are explained. Finally, future work is described.

A. Experiment Plan using GENI Wireless Testbed

The system is planned to be developed and tested by conducting experiments using GENI testbeds. GENI offers a national facility that supporting exploration designs for a future global networking infrastructure [11], which provides a great opportunity to test our proposed system using wired and wireless testbeds.

Figure 3. Clemson GENI Wi-Fi and WiMAX testbeds

Among one of the major campuses involving in GENI projects, Clemson developed and deployed an outdoor Wi-Fi mesh network and WiMAX network as a part of GENI wireless infrastructures. There are 13 Wi-Fi APs (PC Engine-based devices using IEEE 802.11b/g protocol) installed on road-side light poles and one WiMAX base station (Airspan) installed on the roof of the highest student dormitory in the campus, as marked in Fig. 3. These Wi-Fi APs are within the range of the Clemson WiMAX base station. This overlapping coverage area provides an opportunity to test our handoff solution between two wireless access technologies.

Both types of wireless infrastructures are connected to OpenFlow enabled HP switches through VLAN connectivity. OpenFlow provides the possibility and convenience for centralized controlling in the networks. A private VLAN at Clemson could be created and attached to all the switch ports connecting wireless infrastructures. A database server will also be attached on this VLAN. A single Floodlight controller [12] is used to control the wired network.

B. Database and On-board Device

The database will leverage a broadband wireless mapping project at Clemson University referred to as CyberTiger [13]. CyberTiger provides a measurement infrastructure for assessing the coverage and performance of heterogeneous wireless systems. Students download and run Android or iPhone applications (i.e., the client) that interact with performance servers. The client obtains data points that include the time, the GPS location, the handset make and operating system version, the active radio interface, the wireless operator, the signal strength, and the results of the performance assessment that has been selected[1]. The possible performance assessment includes an IP-based ping, a UDP or TCP throughput test or a video streaming test. The client device periodically obtains data points. The data points are immediately transferred to the CyberTiger database. If that is not possible (i.e., IP connectivity was lost in that location), the data sample is entered in the local device database. Once the device re-establishes IP connectivity, the contents of the local database is transferred to the CyberTiger database. The intent of the project is to engage the student population and ‘crowdsource’ an accurate coverage map. The CyberTiger site includes a visualization web interface allowing users to, for example, see where WiFi exists or to compare the coverage of different cellular operators on campus.

[1]Unfortunately, Apple has removed the library that provided radio interface information on an iPhone platform. Therefore the iPhone client cannot report the signal strength.

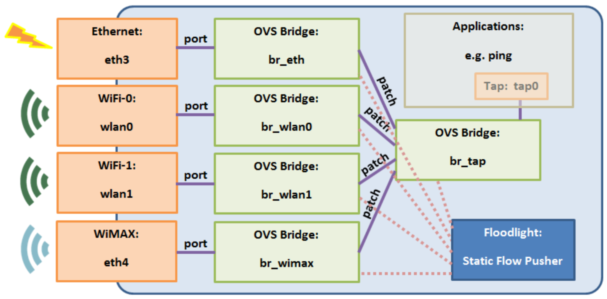

Figure 4. PC Engine Open vSwitch configuration

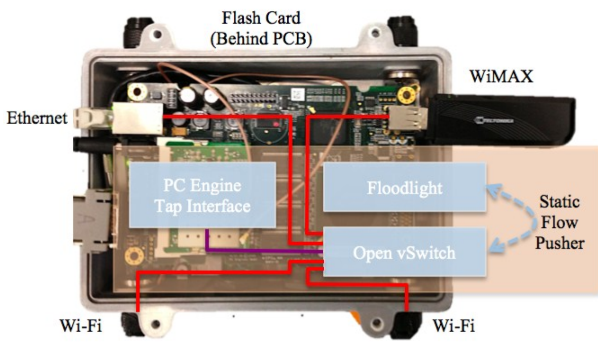

Figure 5. PC Engine software and hardware

The on-board device is being developed on a PC Engine [14] running Debian Linux. We have chosen to run the software package Open vSwitch [15], combined with the Floodlight controller, to leverage the ability to create a virtual OpenFlow-controlled switches on-board the PC Engine. On these virtual switches, the physical interfaces of the PC Engine are connected to Open vSwitch, and a virtual interface is also connected. This Open vSwitch software configuration is depicted in Fig. 4. The end-user will use a default route set on the PC Engine to the virtual interface for all network connections, while the handoff execution will handle which physical interface to use. The software and hardware setup is shown in Fig. 5. The Static Flow Pusher feature of Floodlight allows for flows to be inserted manually, as determined by the handoff decision. These flows rewrite the source and destination IP/MAC addresses to forward packets into and out of the interface chosen by the handoff decision. A Python script leverages the Static Flow Pusher API to add and remove flows. In addition, using Floodlight's module loading system, a custom module will be configured to perform ARP packet rewrite based on the IP address of the physical interface chosen by the handoff decision.

C. *Handoff Performance on PC Engine Device*

The vertical handoff execution was implemented between two different technologies onboard the PC Engine – Ethernet and WiFi – using the OpenFlow solution proposed in the HaaS architecture. The PC Engine serves as the client, while a PC serves as the external server. Although it is not a wireless technology, Ethernet was chosen to demonstrate a significant bandwidth difference (necessitating a handoff) between itself and WiFi. However, this solution supports any IP interface, including WiMAX, as depicted in Figures 1 and 2.

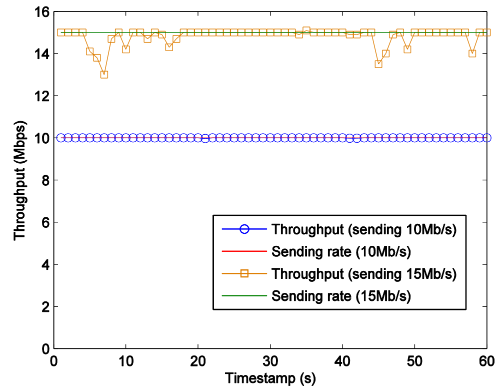

Figure 6. Throughput measurements

Ping tests of the demonstration's topology reveal an average vertical handoff time of approximately 2.43 milliseconds when switching between Ethernet to WiFi (and vice versa) for 16 runs.

Iperf tests with a specified UDP traffic sending rate of 10 Mb/s and 15 Mb/s on the demonstration's topology show a case where a handoff between WiFi and Ethernet is necessary to achieve stable throughput as shown in Fig. 6. In both cases, handoffs were executed from WiFi to Ethernet at 20 second and from Ethernet to WiFi at 40 second. For 10 Mb/s case, the throughput is perfectly stable before and after handoffs. For 15 Mb/s case, disturbance of throughput when using WiFi connection (0~20 sec and 40~60 sec) occurs due to the sending rate is too fast such that some packets were dropped in the buffer because of the transmission rate limitation in the wireless channel. Results from Fig. 6 indicate that the throughput performance is stable before and after handoffs without any major effects.

D. *Future Work*

The future work contains three aspects: (1) Building the database. The implementation of the structure for the database is yet to be finished. The preliminary data collection is also needed. (2) The development of applications exchanging messages between database server and vehicles/RSUs is required. The applications take charge of update messages from vehicles/RSUs and handoff decisions pushing from server to vehicles. (3) The development of on-board devices is also necessary. Currently, the PC Engine device is able to obtain individual IP addresses on each of the interfaces- WiMAX and WiFi. Using Open vSwitch and a Python script, we can switch between these interfaces on a time interval and send and receive data on each network via the ping utility and iperf, for example. These utilities are directed into the virtual network via Open vSwitch and a default route set as the virtual interface. However, the success is partially attributed to the manual addition of ARP table entries at each hop in the path. Currently, the OpenFlow standard does not support ARP packet rewrite; however, Floodlight supports filtering and modification of ARP packets. The Floodlight controller will need to perform packet rewrite for ARP packets to match the IP and MAC address of the physical interface chosen by the handoff decision. This will be implemented by writing a custom Floodlight module and compiling it with the Floodlight controller.

IV. SUMMARY

This paper proposes a handoff-as-a-service architecture for optimal handoff strategies in vehicular networks. By establishing and maintaining a database, the HaaS service computes optimal handoff decisions based on network properties for both moving vehicles and RSUs in heterogeneous networks. Handoff decisions are made considering the three metric classes of availability, performance, and pricing. The server pushes the handoff decisions to desired vehicles, and the on-board device eventually executes the handoff based on the decision received. The wired links between the server and wireless infrastructures and handoff executions are controlled by OpenFlow technology. Experiments on PC Engine show the delay performance is good and throughput performance is stable before and after handoff. Future development work and experiments using GENI testbeds are planned.

ACKNOWLEDGMENT

The authors are supported in part by National Science Foundation grant CNS-1049947, ECCS-0948132 and GENI subcontracts1833A and 1843C.

REFERENCES

[1] V. Brik, A. Mishra, and S. Banerjee, "Eliminating handoff latencies in 802.11 WLANs using multiple radios: applications, experience, and evaluation," in *Proc. ACM SIGCOMM 2005*, Philadelphia, PA, USA, 2005, pp. 27-27.

[2] K.-K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, and N. McKeown, "The stanford openroads deployment," in *Proc. ACM WiNTECH 2009*, Beijing, China, 2009, pp. 59-66.

[3] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," in *Proc. IEEE INFOCOM*, Miami, FL, USA, 2005, vol. 1, pp. 675–684.

[4] T.-Y. Wu, W.-T. Lee, F.-H. Liu, H.-L. Chan, and T.-H. Lin, "An efficient pre-scanning scheme for handoff in cooperative vehicular networks," in *Proc. IEEE PIMRC 2011*, Toronto, Canada, 2011, pp. 583-587.

[5] E. Stevens-Navarro, and V. W. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks," in *Proc. IEEE VTC 2006 Spring*, Melbourne, Australia, 2006, vol. 2, pp. 947-951.

[6] B.-J. Chang, and J.-F. Chen, "Cross-layer-based adaptive vertical handoff with predictive RSS in heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3679-3692, 2008.

[7] K. Shafiee, A. Attar, and V. C. Leung, "Optimal distributed vertical handoff strategies in vehicular heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 534-544, 2011.

[8] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for DSRC vehicle-to-vehicle Communiation," in *Proc. ACM VANET 2004*, Philadelphia, PA, US, 2004, pp. 88-88.

[9] PD TIS. [online]. 2013, http://www.tis.army.mil/products.html (Accessed: 29 Jan 2013)

[10] CAMP VSC, "Vehicle Safety Communications Project Task 3 Final Report - Identify Intelligent Vehicle Safety Applications Enabled by DSRC," USDOT, 2005.

[11] GENI. [online]. 2013, http://www.geni.net/ (Accessed: 30 Jan 2013)

[12] Floodlight Openflow Controller. [online]. 2013, http://floodlight.openflowhub.org/ (Accessed: 30 Jan 2013)

[13] CyberTiger. 2013, http://cybertiger.clemson.edu/ (Accessed: 30 Jan 2013)

[14] PC Engines. 2013, http://www.pcengines.ch/ (Accessed: 30 Jan 2013)

[15] Open vSwitch. 2013, http://openvswitch.org/ (Accessed: 30 Jan 2013)