VANESS: DNS for Nomadic Users in Vehicular Networks

Paulo Santos
DETI
University of Aveiro
Campus Universitário de Santiago
3810-193 Aveiro, Portugal
pjcs@ua.pt

Susana Sargento
Instituto de Telecomunicações, DETI
University of Aveiro
Campus Universitário de Santiago
3810-193 Aveiro, Portugal
susana@ua.pt

José Maria Fernandes
IEETA, DETI
University of Aveiro
Campus Universitário de Santiago
3810-193 Aveiro, Portugal
jfernan@ua.pt

ABSTRACT
In this paper we propose VANESS, a naming service that supports user-to-user communication on mobile devices within a heterogeneous scenario comprising typical Internet Service providers and ad-hoc networks formed by vehicles and roadside units, VANETs. VANESS provides real-time mapping between users and their current network endpoints, which are vehicle's nodes on the VANET or an Internet address otherwise. VANESS was fully implemented on Android OS with results proving its feasibility, and on iOS, showing its multi-platform capabilities.

Categories and Subject Descriptors
[Networks] Naming and addressing; [Information systems applications] Spatial-temporal systems; [Applied computing], Event-driven architectures

General Terms
Algorithms, Management, Design, Experimentation,.

Keywords
Vehicular network, user-to-user communication, REINVENT, gateway, naming services, Android.

1. INTRODUCTION
Vehicular ad-hoc networks (VANETs) have been attracting big interest both from researchers and from the automotive industry. VANET nodes are vehicles or roadside units that communicate with each other, when within the range, and relay messages between them so that messages reach their destination.

This paper proposes VANESS, a Domain Name System (DNS) for nomadic users in vehicular networks: a naming service that can be used as a resource to support user-to-user and user-to-business communication within a heterogeneous scenario comprising typical ISPs (Internet Service Provider) and VANETs focused on mobile devices. VANESS is tested in simplified and controlled scenarios, as well as in real test scenarios using two on-board units in vehicles.

2. VANESS ARCHITECTURE
VANESS is deployed over a typical VANET and involves a set of on-board units (OBUs) and road-site units (RSUs). OBUs are placed on vehicles enabling them to communicate with other vehicles/OBUs in an ad-hoc form using IEEE 802.11p standard (WAVE) [1], and integrate other devices like smartphones through common Wi-Fi technology. RSUs may also serve as outside gateways (Figure 1).

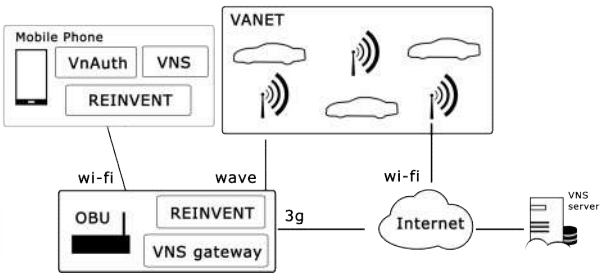

Figure 1. VANESS architecture: The phone has modules to manage accounts, communications and the naming service; The OBU has proxy modules for communication between the phone and the vehicular network (REINVENT), and the phone or vehicular network to the internet (gateway).

VANESS builds on top of REINVENT [2], a solution installed both in the smartphones and on the OBUs that supports the integration of mobile applications and vehicular networks. VANESS introduces two modules: VNS, the local vehicular naming service, and VANET Authentication (VnAuth).

VnAuth handles the users' authentication (sign-in/sign-up/login/logout) and vehicle exchanges, reporting them to VNS – similar to the Social Sign-in model, a device-centralized login and sign-up method.

The VNS on-device module is responsible for keeping track of the users' alias and associated VANET addresses, i.e., the vehicle IDs that are the current gateway to the VANETs. Whenever a user logs in a node in the VANET using REINVENT, VNS is notified creating a new VNS entry. This change is then propagated, whenever possible, by VANESS to the other VANET nodes and to the VNS server. VNS gateway translates VANET format to HTTP requests, and converts the response back from HTTP to the VANET format. Each VNS message or table entry is associated with the timestamp representing the time when the connection is made (entering a vehicle, signing-in or sign-up). This timestamp is used to distinguish outdated entries from new ones.

2.1 Querying Protocol
For VANESS we opted to extend the existing REINVENT API, allowing messages to be sent either to a vehicle ID or to a user alias.

MOBIQUITOUS 2015, July 22-24, Coimbra, Portugal

DOI 10.4108/eai.22-7-2015.2260262

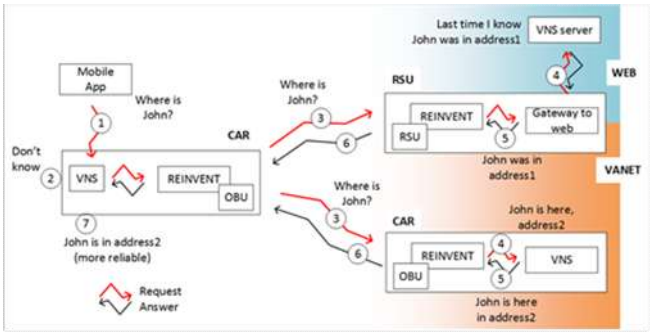

Figure 2. VANESS protocol.

Each time any application sends a message to a user alias, REINVENT queries the local VNS for a translation as depicted in Figure 2.

After receiving a request (1), the local VNS broadcasts to other nodes (3) a request for that alias translation. VNS allows the option of using cached values to improve performance, answering with its local registry (2). The VANET requests will reach other car's OBUs or an RSU connected to the web (3). Each phone using VANESS, when receiving a translation request, will compare the alias from the request with the user(s) logged on the phone. If the alias matches the request, this means that the user is logged in the current device and it sends a response for that alias to the VANET with its own vehicle ID – in the example, it successfully arrived to the car of John, for which we asked the address (4).

If the request reaches a node with Internet connection, namely an RSU, these requests are sent to the centralized VNS server that will respond with its last known association (5). With the server's answer, the translation process does not stop even if the destined OBU is not available at that moment. This is important due to the VANET's frequent topology changes. Regardless of the possibility of both the server and the destined OBU answer to the VNS request (6), the VNS will assume that the first incoming answer is correct and use it to address the message. This avoids waiting periods for a second response that may never come. If a more recent translation arrives with a different mapping, then VNS will re-send the original message again to this new vehicle ID on the assumption that the first translation was wrong or outdated – as in the Figure 2 in (7). This method has a small impact in the network's traffic caused by the lost messages, but makes the overall process faster in most transactions, avoiding the use of a translation's response timeouts both for the server and VANET response.

3. VNRIDE: PROOF OF CONCEPT

We tested VANESS with a proof of concept application, VNRide, which provides a chat allowing message exchange messages via VANET relying on the VNS to perform alias to network addresses translation. We used a scenario with two OBUs with IEEE 802.11p and Wi-Fi technologies, and one mobile phone connected at each one, as depicted in Figure 3. We consider that each OBU is connected to a different vehicle. We run several tests configurations from direct VANETs message exchange through intermediated scenarios passing through VANET nodes and/or VNS web server.

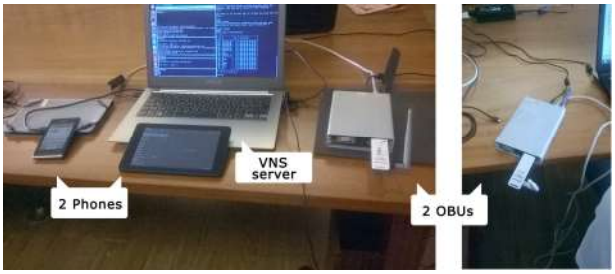

Figure 3. Setup used for tests. Two OBUs with one mobile device connected to each one and a computer running as the VNS server and sharing Internet connection to the OBU.

After these tests, we concluded that VANESS is a feasible and scalable prototype, which simplifies the development of user-level applications, since it removes the endeavor of user discovery for each new mobile application that needs it. Although it does not provide a delivery guarantee if only IEEE 802.11p is supported (and cellular is not used), it is scalable due to its reduced number and size of message exchanged.

4. CONCLUSIONS

VANESS provides a naming service in VANET environments within mobile applications that can be used to support user-to-user communication to third-party applications on vehicular networks – in Android applications it only needs to use a "Login Button" supplied by VANESS - a device-centralized authentication process - as a quick setup to start using VANESS's API.

VANESS has been successfully tested with two real on-board units and a chat application. Since the system has a centralized component (VNS server) able to have information about all vehicles independently of the ad-hoc networks he is in (when that network has a reachable gateway), it would be useful the creation of a proxy between those VANETs, enabling messages to be sent to a completely different ad-hoc network in the world, if each had at least one gateway available.

5. ACKNOWLEDGMENTS

This work was partially funded by the IT Internal Project SenseBusNet (UID/EEA/50008/2013), by the EU Capacities Futures Cities project (http://futurecities.up.pt/site/) and by National Funds through FCT - Foundation for Science and Technology PTDC/EEI-ELC/2760/2012, CMUP-ERI/FIA/0031/2013, Pest-OE/EEI/LA008/13, PEst-OE/EEI/UI0127/2014, UID/EEA/50008/2013 and Incentivo/EEI/UI0127/2014.

6. REFERENCES

[1] IEEE 802.11p standard, http://standards.ieee.org/findstds/standard/802.11p-2010.html, 2010

[2] F . Oliveira, S. Sargento, J. Fernandes, A. Cardote, "REINVENT: Accessing Vehicular Networks in Mobile Applications", IEEE Int. Symposium Computer and Communications (ISCC), pp 23-26, June 2014.