A NetBSD-based IPv6/NEMO Mobile Router

Jean Lorchat, Koshiro Mitsuya
Keio University
Graduate School of Media and Governance
Fujisawa, Kanagawa 252-8520, Japan
Email: lorchat,mitsuya@sfc.wide.ad.jp

Romain Kuntz
The University of Tokyo
Gr. School of Information Science and Technology
Information and Communication Engineering
Email: kuntz@sfc.wide.ad.jp

Abstract— **This paper defines the problem statement of vehicle-embedded networking in order to communicate with the infrastructure (the Internet) as well as with other cars. Based on this problem statement, we explain the steps that allowed us to build a mobile router addressing this problem by using state of the art software. This software includes the NetBSD-current kernel and networking code developed by the Japan-based WIDE project working groups: the KAME IPv6 stack with SHISA extensions for Mobile IPv6 (MIPv6) and Network Mobility (NEMO) support, and the Zebra-based OLSR daemon with IPv6 extensions allowing for a Mobile Ad Hoc Networks (MANET) and NEMO cooperation, formerly known as MANEMO.**

I. Introduction

Current research on Intelligent Transportation System (ITS) focuses on vehicle-to-vehicle communication and information exchange between vehicles and the infrastructure network (Internet). In the near future, vehicles will embed various computers and sensor nodes, making a network, whose data will be exchanged with nodes in the Internet for various purposes such as monitoring, safety, or entertainment. For that purpose, the CALM[1] (Communications, Air Interface, Long and Medium Range) architecture specified at ISO (TC204, WG16) recommends the usage of IPv6 and IPv6 mobility protocols to ensure permanent and uninterrupted communication while moving in the Internet topology.

NEMO Basic Support [1], specified at the IETF in the NEMO Working Group [2], was standardized to solve the IPv6 network mobility problem. NEMO Basic Support allows a group of nodes to connect to the Internet via a gateway: the mobile router. It can change its point of attachment to the IPv6 Internet infrastructure while maintaining all the current connections transparently for all the nodes within the mobile network and their correspondent nodes. The only node in the mobile network that is managing the mobility is the mobile router itself. It updates its current location in the Internet to a special router known as the home agent, which is located in the home network. The home agent maintains a table of relationships between mobile routers permanent addresses, temporary addresses, and mobile network prefixes. All the traffic to or from the mobile network is exchanged between the mobile router and the home agent through an IPv6-over-IPv6 tunnel. This protocol can then be used to bring global access to any car component by adding a mobile router to the car environment. This is one of our goals in the InternetCAR project [3], and we show our communication model on Fig. 1.

NEMO Basic Support is thus a very likely architecture to ensure permanent connectivity in mobile environments. Although the main use case would be to connect any kind of vehicles to the Internet (such as cars, trains, buses, etc.), the underlying architecture would be the same (IPv6 and NEMO Basic Support) but customized at the application layer according to the usages: monitoring, safety, entertainment, etc.

The Internet Connected Automobile Research (InternetCAR) Working Group was established within the WIDE Project [4] since 1998 to con-

[1]http://www.calm.hu/

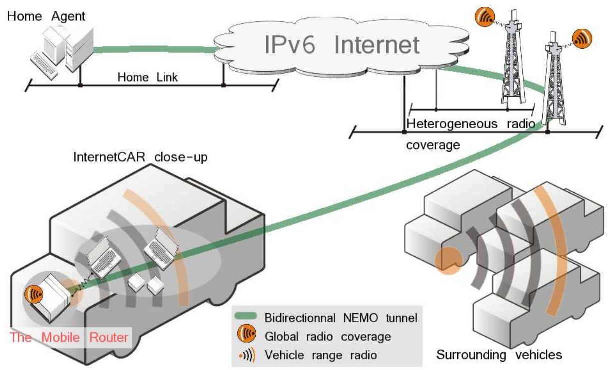


Fig. 1. InternetCAR communication model

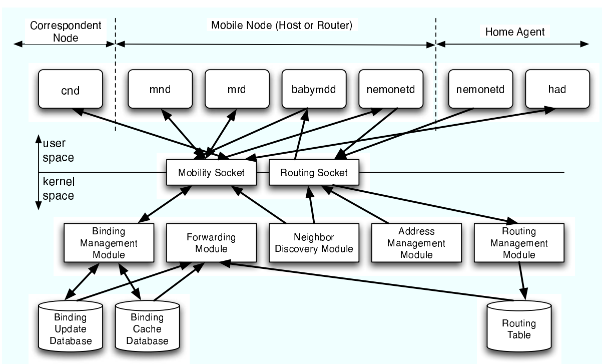


Fig. 2. SHISA daemons architecture

nect vehicles to the Internet [5], [6], [7] by developing the necessary tools and demonstrate their applicability in a real-life testbed. We aim to implement all the missing protocols to build an in-vehicle Mobile Router. It is designed to support IPv6 mobility and multihoming (simultaneous usage of several interfaces to share and load-balance the traffic, or for fault-tolerance purposes).

InternetCAR is also developing a monitoring application for demonstration purposes. Most of the outputs are freely available implementations for BSD operating systems, such as contribution to the KAME IPv6 stack [8], and the SHISA mobility stack [9].

The previous in-vehicle mobile router was based on the NetBSD 1.6.2 Operating System. The KAME IPv6 stack and SHISA stack [10] were used to manage the network mobility. Basic multihoming features allowed vertical handover between Wireless LAN and cellular interfaces. However, no vehicle-to-vehicle communication is possible, and no monitoring software on the mobile router makes the evaluation of the system difficult.

The purpose of this paper is twofold: first we make out and explain the constraints of such in-vehicle network architecture for both hardware and software sides. We then explain the work done on the implementation side to build a new version of the InternetCAR's in-vehicle mobile router, based on the NetBSD operating system with respect to the previously defined constraints. We then show an evaluation of this work.

II. A CONSTRAINED ENVIRONMENT

As explained in the previous section, one goal of the InternetCAR project is to build an embedded mobile router that is suitable for car operation. And while the car environment is not as constraining as some other environments from the embedded computing domain, it still has some features that must be accounted for when choosing hardware and software solutions to specific problems. We will explain these environment peculiarities by distinguishing between hardware and software considerations.

A. Hardware related considerations

Since the car is moving, we have to pay extra attention to the toughness of the hardware. The road conditions can make the embedded computer bump, so that moving parts should be avoided as much as possible. And of course, as long as the car is moving, it makes wireless communications (and especially all kinds of radio communications) mandatory.

As an embedded car equipment, the size of the mobile router must be kept to a minimum in order to save space and weight inside the car. This means avoiding traditional personal computer design and looking for single board computer alternatives.

Although AC current from the mains is not directly available inside the car, we have sufficient access to power resources through the car power source, either the alternator or the car battery.

Eventually, cars are equipped with lots of cable, yet there might be no networking cables available to interconnect equipments within the car, especially for current vehicles. Which means that we might have to resort to wireless communications again to provide in-car networking.

B. Networking related considerations

We have already introduced the ISO specified recommendation about IPv6 and NEMO protocols. These protocols allow to hide mobility in a very convenient way, especially by relieving car equipments from any mobility-related overhead, handled by the mobile router.

However, in addition to these next generation mobility protocols, we need to take the vehicle to vehicle (V2V) scheme into account. Using the previously mentioned mobility protocols, V2V communication can become very painful because the traffic has to go all the way back to both home agents. In this case, protocols for mobile ad hoc networks can be used to discover neighboring mobile routers and to route traffic between them.

This means that several daemons are going to be responsible for route injection (i.e. the mobility daemons and the ad hoc network daemons) and it is mandatory that they can collaborate in an efficient way. Above all, since mobility daemons are going to define the default route through the NEMO tunnel, this must be done in the regular way that allows longer prefixes to retain priority during the routing process.

C. Problem Statement

From the previous observations, we define the following problem statement: "*How to provide optimized connectivity to a moving car environment and all its attached equipment and users ?*". Keeping in mind that the recommended architecture from ISO must be followed to maintain compatibility with future solutions, but slightly amended to optimize V2V communication. And while these software goals must be achieved, we must also obey the guidelines that come from hardware constraints of the car environment.

III. INTERNETCAR TEAM IMPLEMENTATION

A. Hardware Platform

By taking all hardware-related constraints into account, we decided to use a Soekris[2] single-board computer as the key element for our mobile router. While it features an integrated processor, memory and two network controllers on the same board, it also provides room fox expansion with two Cardbus and one MiniPCI slots. This will allow us to use many wireless network interfaces.

The mass storage is fitted through a Compact Flash slot, which can accommodate either a flash card or a micro hard-disk. We chose to use flash cards to achieve a zero-spin architecture, which means that our mobile router has no moving part.

B. Operating System Choice

We chose the NetBSD operating system for several reasons. The first being the portability, which guarantees that switching from the current hardware platform to another will be easier. And since we wanted to use MiniPCI wireless network interfaces, we decided to use recent NetBSD kernels, as compared to the previous version of the mobile router.

NetBSD-Current (as of mid-October) was installed on the compact flash card using a laptop. Then, we proceeded to the customization of the root filesystem. This included removing system services that are not useful for our embedded operation, while adding specific services related to networking stack and monitoring (see III-C to III-F).

Then we modified the mountpoints so that the root filesystem is mounted read-only (to protect the flash storage) while a memory filesystem is mounted in frequently written locations like `/var/log`, `/var/run` and `/tmp`.

[2]http://www.soekris.com

C. Mobility protocols

As Mobile IPv6 and NEMO Basic Support implementation, SHISA is freely available for BSD variants. However, SHISA is an extension of KAME IPv6 implementation and it is not available for any BSD main branch yet. We thus have ported SHISA to NetBSD-Current with the help of the SHISA developers team.

The main design principle of SHISA is the separation of signaling and forwarding functions. The operation of Mobile IPv6 and NEMO Basic Support is basically IP packet routing (forwarding or tunneling). In order to obtain better performance, the packet routing should be done in the kernel space. The signal processing should be done in the user space, since the process is complex and it is easier to modify/update user space programs than kernel. This separation provides both good performance and efficiency in developing the stack.

In SHISA, a mobile node (host or router) consists of small kernel extensions to process mobility headers and to forward packets, and several user land daemons (MND, MRD, BABYMDD, MDD, NEMONETD, HAD and CND). MND/MRD are daemons which manage bindings on a mobile node. BABYMDD/MDD are daemons which detect the changes of temporary addresses and notifies them to MND or MRD. NEMONETD is a daemon which manages bi-directional tunnels. HAD is a daemon which manages bindings on a home agent. CND is a daemon which manages bindings on a correspondent node. Depending on the node type, one or several SHISA daemons run on a node. For instance, MRD, NEMONETD and MDD are running on our mobile router. The relationship between user land daemons and kernel is detailed in Fig. 2.

D. Mobile Ad Hoc Network protocol

To address the efficiency problem of V2V communication, we explained in section II-B that it would be possible to route traffic using a Mobile Ad Hoc Network (MANET) protocol. We decided to base our work on a proactive routing protocol because the neighboring nodes discovery process is more tightly integrated by distributing topology information. The protocol we chose is OLSR [11].

Although the RFC does include enhancements to announce host/network associations, it is not definite about IPv6 networks nor IPv6 hosts. We use a routing daemon developed in Keio University and functioning as a plug-in to the Zebra framework. It is already going beyond the scope of the RFC by supporting IPv6 hosts.

We modified this daemon to support IPv6 mobile routers announces (i.e. which prefix is reachable through which node) and added support for sub-second precision in Zebra timers. It is then possible to discover mobile routers that are nodes of the MANET at the current time, and use OLSR routing services to route V2V traffic. This is further described in [12] where we show that it is possible to discover new nodes and broken links in less than 100 milliseconds.

E. Interface management

Since an automobile moves around, the mobile router in the automobile needs to perform handovers between wireless access points. For this operation, it is required to carry out discovery of base stations, to connect to one of these stations, and to detach from the previous base station.

Casanova is an automatic Wireless LAN SSID/Wepkey switcher for FreeBSD and NetBSD. Casanova handles ESSID and wepkey and configures the wireless network interface while handing over from one access point to another. Casanova also acts as an IPv6 address manager. It requests to configure a new IPv6 address when the mobile router is connected to a new link, and to remove an old IPv6 address when disconnected.

F. Monitoring software

The embedded nature of the mobile router requires a flawless operation. However, there are some cases where mobile router operation is not possible, especially when no connection is available. This can be reported by our dedicated monitoring software. There are two small daemons sharing the same codebase. One is showing statistics locally using the front LED of the Soekris box (to diagnose connectivity problems). The other one is the SONAR[3] client, which records and prepares the data for transmission to a remote repository. It can be stored in the filesystem too, and sent later so as not to disturb ongoing experiments.

The monitoring architecture is very modular and allows for fast development and integration of new features to monitor. It is written in C and supports several platforms: NetBSD, FreeBSD and Linux. Statistics are polled on a regular basis using a user-defined interval, between a few milliseconds and several hours. This requires interaction with routing daemons (SHISA and OLSR) and the kernel.

A report is built for each polling interval using an XML tree structure. These reports can be sent regularly (another user-defined interval) or even kept in a local storage area and collected later. For an in-depth presentation of the monitoring architecture, please refer to [13].

IV. Evaluation of implementation

The mobile router that we described in this paper was implemented during the October month in year 2006. It can be seen on Fig. 3.

The picture on Fig. 3(a) shows the single-board computer (Soekris net4521) that we use to implement our mobile router architecture. Both pictures on Fig. 3(b) and Fig. 3(c) show experiments that we made using the mobile router. The former is an electric car manufactured by Toyota which has all approvals required to be driven on open road. The mobile router can be seen just behind the driver's seat. The latter illustrates the experiment led at the Open Research Forum in Tokyo (October 2006), where we equipped an electric bus with the mobile router, SNMP sensors and an IPv6 camera.

Before the implementation process, we were facing some questions about the behavior of networking component with respect to the each others. However, the interaction between the SHISA/KAME stack and the Zebra daemons went very smoothly. In fact, the main problem we faced during the implementation was a routing issue when using the KAME stack : at that time, it was impossible to send messages from SHISA daemons through a gif tunnel because the endpoints were not recorded in the neighbor cache.

With this achievement, we could perform some measurements on the workbench before the mobile router is put into vehicles. We especially investigated the booting time for the whole system and the amount of time required before the mobile router is successfully registered to its Home Agent, or before the network interface becomes available. These results are shown in table I.

Following this implementation, several software releases will happen. The casanova program used for interface management has already been released[4]. The SONAR client used for statistics monitoring is due to be released by the end of the year. Although we used a NetBSD-current port of the SHISA stack for NetBSD-current, we can not tell for sure when it is going to be released to the public.

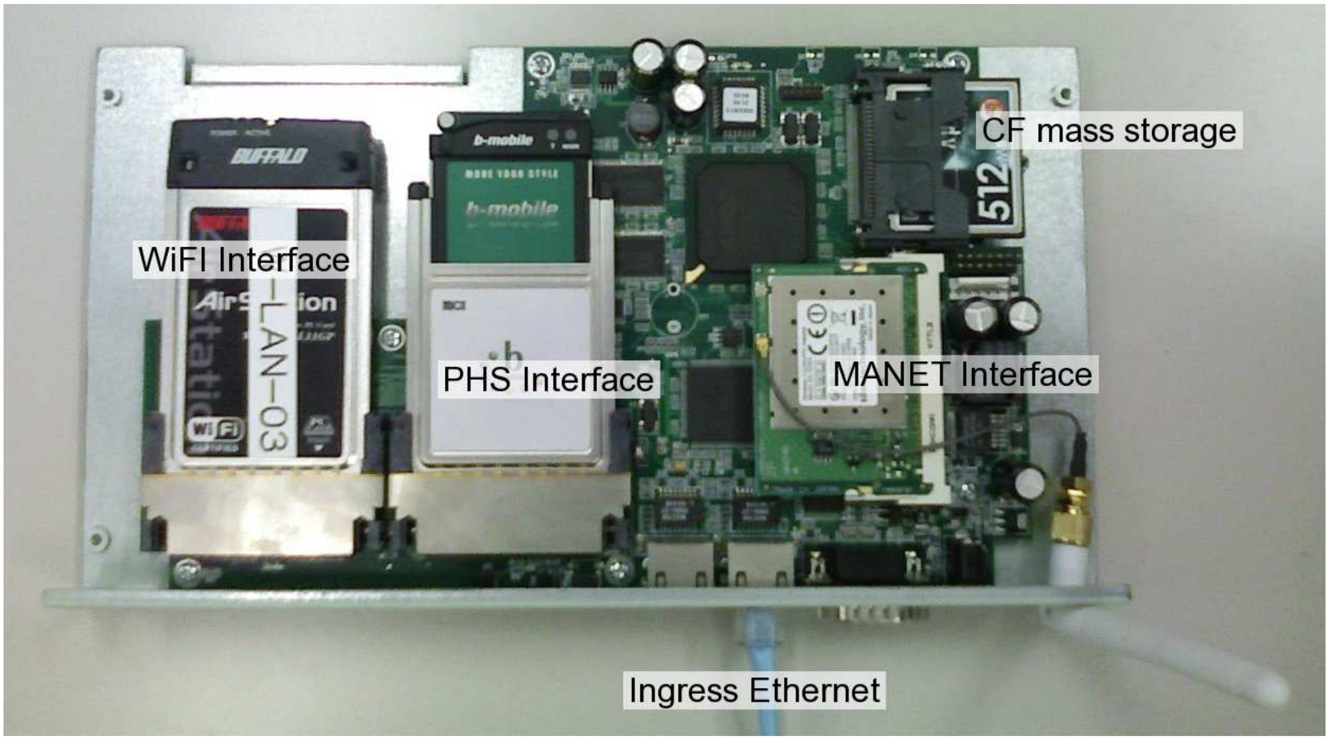
V. Conclusion

After a thorough description of the target environment and its constraints, we could define a specific problem statement that led us to the current implementation of a in-vehicle mobile router. It is based on the NetBSD-current (October 2006) operating system and uses several advanced networking code from the WIDE project.

The core IPv6 and NEMO functionnality is supported by the KAME/SHISA networking stack, while MANET routing is handled by the OLSR routing protocol, using a modified implementation built as a plugin to the Zebra framework. These routing daemons were made to cooperate and routing decisions are based on the availability of mobile routers in the MANET vicinity. On a lower level, we developped software to control the network interfaces used by the SHISA stack. Eventually, monitoring is made by

[3]http://sonar.nautilus6.org

[4]http://software.nautilus6.org



(a) The Soekris SBC

(b) A test vehicle : Toyota COMS

(c) The ORF experiment : Marunouchi Shuttle

Fig. 3. The mobile router implementation

	Total boot time	NetBSD boot	Interface startup	MIPv6 / NEMO registration
PHS	79 s	60 s	11 s	8 s
WIFI	77 s	58 s	11 s	8 s

TABLE I
BOOTING TIME PROFILING

specific software that polls every component that needs to be reported.

Using this implementation, we were able to show preliminary results about booting time that confirm the fact that the choice of the NetBSD OS is suitable for in-car operation (with one minute and a half booting time).

As a next step, we are going to use this mobile router on a daily basis in personal cars, and at specific exhibitions like the ORF 2006 in Tokyo [5]. However, to be operable at public scale, we need to further investigate security issues at the network and datalink layers. The former will be achieved using IPSec transport between the mobile router and the home agent while we might resort to hardware encryption (WEP or WPA) for the latter.

ACKNOWLEDGMENTS

We would like to thank the people from SHISA team for helping us with NetBSD-current stack, and all the people who got involved in this project at any given time, especially Keiichi Shima, Keisuke Uehara and Manabu Tsukada.

REFERENCES

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," IETF, Request For Comments 3963, January 2005.
[2] "IETF Network Mobility (NEMO) Working Group Charter," URL, As of September 2004, http://www.ietf.org/html.charters/nemo-charter.html.
[3] "InternetCAR Working Group, WIDE Project," Web page, http://www.icar.wide.ad.jp.
[4] "WIDE Project (Widely Integrated Distributed Environment)," Web page, http://www.wide.ad.jp.
[5] T. Ernst and K. Uehara, "Connecting Automobiles to the Internet," in *ITST: 3rd International Workshop on ITS Telecommunications*, Seoul, South Korea, November 2002.
[6] K. Mitsuya, K. Uehara, and J. Murai, "The In-vehicle Router System to support Network Mobility," *LNCS*, vol. 2662, pp. 633–642, 2003.
[7] T. Ernst, K. Mitsuya, and K. Uehara, "Network mobility from the internetcar perspective," *Journal of Interconnection Networks(JOIN)*, vol. 4, no. 3, pp. 329–344, 2003.
[8] "KAME Working Group, WIDE Project," Web page, http://www.kame.net.
[9] "SHISA: an implementation of Mobile IPv6 within the KAME IPv6 stack," http://www.mobileip.jp/, http://www.kame.net/.
[10] K. Shima, R. Wakikawa, K. Mitsuya, T. Momose, and K. Uehara, "SHISA: The IPv6 Mobility Framework for BSD Operating Systems," in *IPv6 Today - Technology and Deployment (IPv6TD'06). International Academy Research and Industry Association*, Aug 2006.
[11] E. Clausen and E. Jacquet, "Optimized link state routing protocol (OLSR)," Internet Engineering Task Force, RFC 3626, Oct. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3626.txt
[12] J. Lorchat and K. Uehara, "Optimized Inter-Vehicle Communications Using NEMO and MANET," in *V2VCOM2006 : 2nd Vehicle-to-Vehicle Communications Workshop*, San Jose, California, USA, July 2006.
[13] K. Mitsuya, J. Lorchat, T. Noel, and K. Uehara, "SONAR : A Statistics Repository of Mobility Platforms," in *The First International Workshop on Network Mobility (WONEMO)*, Sendai, Japan, Jan 2006.

[5]http://orf.sfc.keio.ac.jp