Towards a Dynamic QoS Management Solution for Mobile Networks based on GNU/Linux Systems

Adapting Already Existing Solutions to Vehicular Environments

Gorka Urquiola, Asier Perallos, Itziar Salaberria, Roberto Carballedo
Deusto Institute of Technology (DeustoTech)
University of Deusto
Bilbao, Spain
{gurquiola, perallos, itziar.salaberria, roberto.carballedo}@deusto.es

***Abstract*—The number of applications used in Intelligent Transportation Systems is growing very quickly. This implies a greater consumption of vehicular network bandwidth hence there could be a high probability of delay of priority requests in this networks. Consequently, an exhaustive control of the bandwidth is needed to provide a Quality of Service according to the demands of certain applications. In this paper, a communications middleware to provide the management of the Quality of Service and prioritize applications' requests on mobile networks is tested. The proposed system, in order to reduce development efforts, has been addressed only reusing and configuring already implemented and tested GNU/Linux based software utilities, originally designed to be used in non-mobile environments.**

Keywords-Vehicle-to-Ground Communications; Quality of Service; Requests Priorization; Virtual Private Network; Queue Disciplines; Linux

I. INTRODUCTION

In transportation systems, is not easy to guarantee continuous communications and a stable available network bandwidth inside the vehicles. Common network configurations used in non-mobile environments, such as the ones used in an office (where static network links are used, continuous communication can be assured using wired and backup links [1], and the service failure probability depends on rare environmental factors and internet service provider quality), are not directly adopted in mobile networks. The reason is that the quality of the communication could be affected by several dynamic factors, such as coverage changes according to the location of the vehicle, data packets losses or event cuts in the communication that may occur.

For such networks, it is usual to adopt vehicle to ground architectures [2], in which it is necessary to maintain the communication between the mobiles and control centre nodes or even the communication between all the mobile nodes.

Moreover, the number of applications used in this kind of mobile environment is growing in an exponential way due to the requirements of the Intelligent Transportation Systems (ITS) [3]. The mobile services offered by the internet service providers are not always capable of providing a suitable bandwidth that meets the needs of such applications. Consequently, an exhaustive control of the bandwidth consumption is needed to provide the Quality of Service (QoS) demanded by certain applications, such as in the case of surveillance video streaming (high bandwidth consumption, low priority) and an alarm trigger (low bandwidth consumption, high priority). In these cases, communication requests must be prioritized or delayed assuring priority to the most relevant data traffic and leaving in background the not critical one [4].

In order to have a greater connectivity and coverage, we could use 3G modems for accessing to the Internet. Instead of developing specific software able to manage the different links, establishing the active channel to use (based on factors such as coverage, availability and bandwidth), we decided to combine existing software tools. The aim is to get an easy to develop and deploy communication solution for mobile (vehicular) environments which is able to manage the QoS of applications in a dynamic way [5].

To achieve this target, a system based on a GNU/Linux distribution, using only free and open-source software tools, has been designed and tested. These software tools have a fairly widespread use, which incurs in having an always updated and well documented system. Thus, we can develop a communication system with a minimum initial investment and whose robustness and fault tolerance is guaranteed by the support and contribution of a community of worldwide developers.

The rest of the paper is organized as follows. In Section II, a brief overview of the state of the art is included. The contributions of the developed communication system are included in Section III. The proposed solution design, including the description of the tools used and the reasons for choosing them, is presented in Section IV. Then, in Section V, the real scenario in which the system has been tested is described. Finally, the results of the tests are analyzed in Section VI and the paper ends with the conclusions and future work.

II. STATE OF THE ART

Transportation companies demand greater efficiency for their systems, therefore, wireless communication technologies are growing in vehicular systems. Also, they are also seeking to provide new information services [6]. For many years, the networks used in transport systems have been formed based on separate islands of physical media and protocols [7]. Currently, the existence of multiple transmission alternatives provides higher communication bandwidths [8], but this does not mean a better performance in regard to interoperability, temporary or reliability properties [9]. Thus, there is an increasing complexity in telematics contexts because of the continued growth of the specific systems and solutions, requiring technologies that enable greater interoperability between these solutions [10].

On the other hand, even if the emphasis in developing wireless networks is on network bandwidth and coverage, the applicability of the communication system will largely depend on their ability to provide sufficient data rates (QoS requirements), considering introduced protocol overhead, packet fragmentation and possible retransmissions.

Therefore, wireless communications applied to mobile environment present several limitations related to coverage and bandwidth that can cause service disruptions. Moreover, wireless stations that need to transmit critical information must deal with wireless stations wishing to transmit less priority traffic.

For the purpose of achieving QoS requirements demanded by services, several communication management and prioritization heuristics [11,12] and mechanisms exist [13-15]. Although existing solutions are mainly focused on network aspects and not in final applications and services, other approaches are focused on optimizing the use of the network technologies according to the type of traffic generated by applications (QoS control). Therefore, there is an open research field that can be tackled from two complementary points of view: (1) QoS requirements management, which involves technology concepts related to the information to transmit, and (2) aspects about network conditions that make possible the transmission of that information (bandwidth, coverage, latency, etc.). The work presented in this paper will explore this first approach.

There are multiple works regarding communications optimization, including traffic prioritization and QoS control. However, these works are usually focused on networks instead of applications or services that use these networks [6,16]. In addition, there are industrial solutions designed to respond to these detected communications needs and challenges in transportation systems [17, 18]. But, neither of these projects establishes a communication system that prioritizes data transmissions dynamically.

III. Technical Contributions

There are three main technical contributions of the proposed vehicular communication system:

- *Network traffic regulation*. The number of applications used in vehicular environments is growing very quickly, which implies a greater consumption of network bandwidth. The regulation of applications network traffic becomes important, as an excessive network bandwidth consumption by a secondary application may cause delays or even data packets losses of a priority application. Thus, this can be a problem for those applications that consume lower bandwidth, but which have a higher transmission priority.
- *Network security*. Security in communications is another aspect to consider as applications may be required to transmit sensitive information between the mobile node and ground centres, such as ticketing information using Near Field Communication (NFC) or contact/contactless SmartCards.
- *Onboard subnet management*. It would be desirable that the onboard communication system had to be able to manage a subnet and serve as a gateway to the control applications located on ground centres. It is a requirement that does not require to run all applications on the same device, allowing the use of additional devices, such as sensors or IP cameras, which act as additional nodes in a subnet.

IV. Communications System Design

The proposed system and the later tests have been addressed in a generic manner, as an assumption of the authors, trying to cover a wide range of use cases. Thus, they do not represent real expectations of specific manufacturers and users. Nevertheless, a specific use case of the proposed solution is train-to-earth railway communications [17]. The train units need to transmit heterogeneous information (different size and urgency). Thus, for example, a train requires that critical positioning data of few kilobytes to be transmitted continuously, while other type of data transmission like video streaming may be heavier but can wait to be transmitted until priority data has been sent. The management of these kinds of communications requires very different priority and QoS treatments that could be addressed by our work.

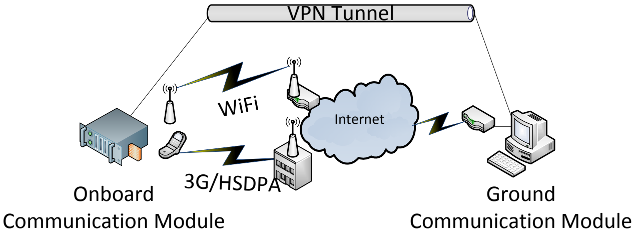

Figure 1. Conceptual architecture of the system.

The proposed communications system follows a vehicle-to-ground architecture based on the existence of an Onboard Communication Module (OCM) and a Ground Communication Module (GCM). The onboard module has two different wireless communication links – a 3G modem and a WiFi antenna – and the ground module has a wired broadband link (Figure 1). The onboard module will only use one of the available links (we refer to it as the active channel). The system will use the WiFi link if a known access point is available, if not, the system will use the 3G modem as network link. A 4G/LTE connection would be another way to implement the mobile connection, but because of the ease of implantation in a wide range of scenarios the 3G option has been finally chosen for the proposed system.

The OCM also has an Ethernet interface in order to manage the onboard subnet. On this subnet other devices can be connected, such as IP cameras, sensors, embedded systems or even PCs or laptops.

Using a host-to-host type Virtual Private Network (VPN) will cover two of the previously presented system contributions. On one hand, the communication will be encrypted with a cryptographic symmetric key, providing an additional security layer for data transmission between onboard and ground modules. Furthermore, the use of a VPN involves the creation and use of a virtual interface whose IP address will be the same regardless of the physical link being used at any time. This means that the use of these virtual interfaces for the communication between the two extremes ensures that applications do not have to change their settings when the active physical channel is changed. This supposes a new abstraction layer for the applications working with this vehicular communications system. Therefore, each of the available mobile nodes will be identified always with the same IP address.

Since most of network traffic will be sent from the mobile nodes to the ground centre, a QoS management middleware will be implemented in the mobile end, setting the network traffic rules on the virtual interface created by the VPN. Thus, it does not matter what the currently active link is, since all network traffic will be transmitted using the virtual interfaces configured on the system.

As the available network bandwidth is not stable as it could be in a static network, the QoS becomes more important. The network consumption priorities should be managed and adapted each time the available bandwidth fluctuates.

Summing, the onboard module must behave like a kind of router able to: manage the host-to-host type VPN to ground module, manage the private subnet of the mobile node, prioritize outgoing network traffic, run third party software and redirect the data traffic of the private subnet to the ground centre.

A. *Software utilities used*

No software was developed in this proposed communications system, but it has tried to combine and configure already existing and available software tools to meet the contributions presented in Section 2.

TABLE I. MATCHING OF SYSTEM CONTRIBUTIONS AND THE SOFTWARE TOOLS USED TO THEIR FULFILMENT

Contribution	**Technical solution and software utility used**
Network traffic regulation (prioritization)	QoS management (iptables + Traffic Control)
Network security	Point-to-point VPN (OpenVPN)
Onboard subnet management	Network gateway (Webmin)

For the deployment of the system, the software used was (Table 1): Ubuntu 11.10 [19] as GNU/Linux distribution, OpenVPN [20] for the host-to-host VPN management and various utilities from the Iproute2 [21] utility collection and Netfilter framework [22], mainly iptables and Traffic Control for the QoS management.

1) *Operating System (GNU/Linux)*

Although it can be found equivalent tools on different operating systems like Microsoft Windows, it was decided to choose a GNU/Linux distribution for two reasons: first, that is free and open source, and second, that is easier to find and modify network management tools than in others.

2) *VPN (OpenVPN)*

As VPN management software, OpenVPN was used due to its ease installation and free use.

A host-to-host type VPN must be configured for each onboard module to be managed from the ground node. The latter is the responsible for managing the communications between the different mobile nodes if they wanted to make a communication from a mobile node to another.

This type of VPN requires that one of the two nodes acts as a server and the other one as a client. Considering that the onboard physical links will have variable IP addresses depending on which the current active link is and the location of the mobile node, the ground module will be the VPN server and will be in charge of receiving request for connection establishment from each of the physical interfaces installed in the onboard modules.

Therefore, the design of the network architecture follows a star topology, where the ground module is the central node of the graph and the mobile modules are the leaf nodes.

The client-server connection establishment is made using the default route defined by the routing table of operating system. This can be modified using the ip route command, from the Iproute2 utility collection [21], available in most of the GNU/Linux distributions. In case of modification of the default route, OpenVPN detects it and manages the reconnection to the server using the new route.

3) *Quality of service (iptables and Traffic Control)*

To ensure the quality of service of the active channel, a combination of iptables and Traffic Control utilities has been used (Figure 2).

Iptables belongs to the framework Netfilter and it is the default firewall used in GNU/Linux. For this solution, its packet marking module will be used. With this module a mark will be added to each data packet redirected to the external network, either from the private subnet or the system itself. This classification is based on the port used to transmit each of the packages, so the data packets of each application running in the mobile node can be classified.

The data packet marking rules are easily configurable and replaceable in case of making changes on the system.

Traffic Control, from the utility collection Iproute2, will manage the queue disciplines, prioritizing the outgoing data traffic. After iptables has marked the data packets according to established rules, Traffic Control associates each mark to a priority class and then classifies and manages the bandwidth usage limits.

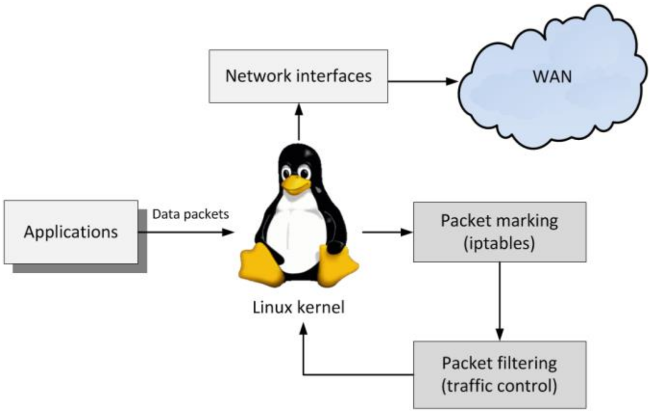

Figure 2. QoS management in GNU/Linux using iptables and Traffic Control utilities.

4) *Queue disciplines (qdisc)*

The queue disciplines determine the way in which data packets are sent. It is important to highlight that it can only be shaped the transmitted data and not the received one. Thus, the OCM will manage the queue disciplines to guarantee the QoS demanded by the applications and the adequate network usage prioritization.

Among the available queue discipline algorithms two have been considered for the proposed system: PRIO and HTB.

The PRIO qdisc (Priority queue discipline) does not need to define the current available bandwidth and it subdivides traffic based on how the Traffic Control filters are configured. It is a strong queue discipline for static networks in which the bandwidth fluctuates, such as neighbor shared network where the interactive traffic and non-interactive traffic must be managed, giving priority to the interactive one.

The Hierarchical Token Bucket (HTB) [23] queue discipline allows dividing the available network bandwidth indicating a maximum and a minimum usage for each application, ensuring that the highest priority applications of the system may have the required bandwidth at any time.

Despite of having to specify the available bandwidth each time the bandwidth changes, the HTB is proposed to be the queue discipline to use. We consider that it is a more adequate queue discipline in mobile networks mainly because it allows to configure the transfer rate per application. Moreover, it can be easily reconfigured with a few commands when the bandwidth changes; therefore, it does a better network prioritization than the PRIO queue discipline.

5) *Gateway configuration (Webmin)*

Ubuntu, as all other GNU/Linux distributions, can be configured to act as a network gateway. However, in order to facilitate this task and to provide a more user-friendly gateway, it was decided to use a web-based interface for system administration. To do this, it was chosen Webmin [24], which has all the necessary features, such as DNS and DHCP server.

The gateway was configured to forward the traffic from the subnet to the VPN tunnel and to apply the previous specified QoS rules in order to shape the network traffic.

V. Tests Set-Up

In order to test our communication system, we have developed a simple application which triggers petitions from an onboard device to the GCM (Figure 3). In this communication, the traffic is forwarded to our system and it is shaped and prioritized according to its predefined configuration. This test application was run in a laptop which was connected to the onboard Ethernet network, so we could test two parts of the system: the network management and network prioritizing method.

It is important to highlight that this simple application is the unique software developed in this project and it has been used only to perform the tests. All the software that composes the solution already existed and was developed by third parties.

Moreover, only for informative purposes, during the tests the geolocation of the vehicle was captured using a standalone GPS device.

To verify that the proposed system works, a test plan has been developed and performed in laboratory settings, using a PC and an embedded system to simulate ground centre and an onboard module. Both systems have Ubuntu 11.10 and OpenVPN installed.

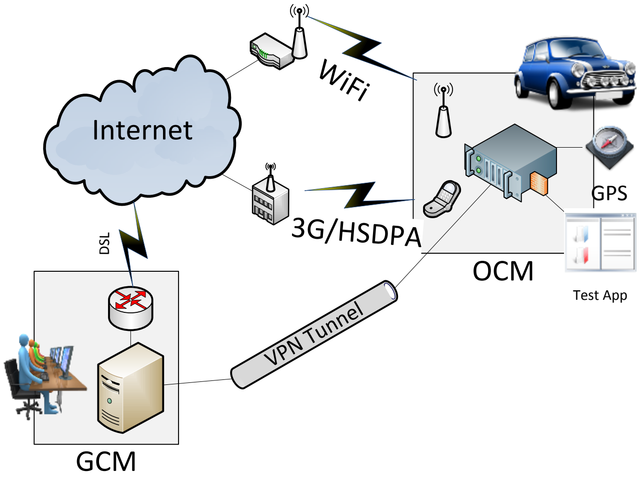

Figure 3. Conceptual diagram of the test implementation.

Summarizing, in this tests some files will be transmitted from the vehicle to ground, each one with different priority. The goal is to validate that the system performs properly. It means that all the requests are transmitted in compliance with the established minimum transfer rate and in the case of having free bandwidth it will be assigned to the highest priority request.

A. *Scenario configuration*

The system was tested in a real vehicular scenario. Although a system like the proposed one actually would be deployed into a public transportation vehicle, the tests were designed to be performed in a private car seeking to emulate the same conditions as would occur in a public bus.

In this section, a description of the geographical scenario in which the tests has been performed is described. There are three main elements in the scenario configuration: the path, the vehicle, and the network link.

In the selection of the scenario path, it was considered to have a bandwidth fluctuating scenario, so a mixed urban and outskirts path was chosen. The path goes from the University of Deusto (Bilbao, Spain) to the beach of Sopelana (Spain), located to 16km away (Figure 4).

As a vehicle, it was used a common private car, a Renault Clio from 2008, and the travel was made in an average velocity of 80Km/h, as it would be in a public bus. In addition, two people were required in the car, one driving and the other one supervising the embedded system and the test application running in a laptop.

Due to the chosen queue discipline behaviour, the system must know the available bandwidth in each moment so that the traffic prioritization is done as intended. For this purpose, before doing the real vehicle travel and test, a current-available-bandwidth-capture was done. Thus, it was used three 3G USB dongle from different Internet Service Providers. The bandwidth data was captured using Iperf [25], a free and open source network tool. Iperf can also provide more data of the network, such as network latency, but for these tests we only needed to use this tool to get the available bandwidth on each moment.

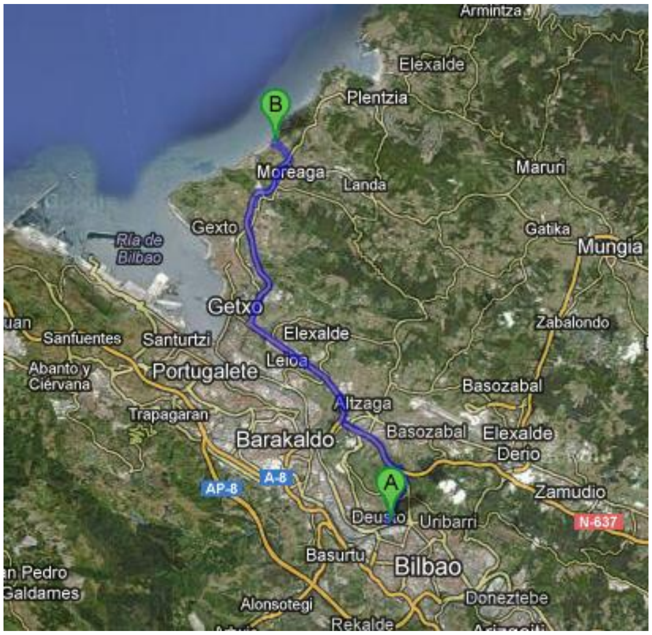

Figure 4. Path of the test scenario.

B. Running the tests

Once the first bandwidth-catching trip was done, the link with more bandwidth changes was chosen. So, the proposed system would be tested in the worse possible scenario to get the best analysis about the proper performing of the communication system.

Having in mind that the system has to know how much bandwidth is available in each moment and assuming that the available bandwidth would be similar to the data captured previously, some scripts were prepared by which the system changed the network prioritization adapting its configuration to the current network status.

Each test was composed of four requests. Each request with a different level of precedence: low, normal, high and priority. The planning of requests (minutes when they are triggered) was the following:

- Minute 0: normal priority request.
- Minute 1: high priority request.
- Minute 2: low priority request.
- Minute 3: the highest priority request.

This test suite was done repeatedly along the path to the end of the trip, so results of different areas can be analysed after the tests execution.

VI. Test Results

To be able to analyse the test results, the developed testing application, every three seconds, logged the transfer rate of each request and a GPS device captured the position of the vehicle. Thus, we could identify the behaviour of the proposed system at any time and location.

Taking the graph showed in Figure 5 as our first test result set, we see that a total bandwidth of 120KB/s is assigned. In this situation, the bandwidth is divided according to the following priority levels and the minimum transfer rates needed:

- The highest PRIORITY request: 70KB/s
- HIGH priority request: 30KB/s
- NORMAL priority request: 15KB/s
- LOW priority request: 5KB/s

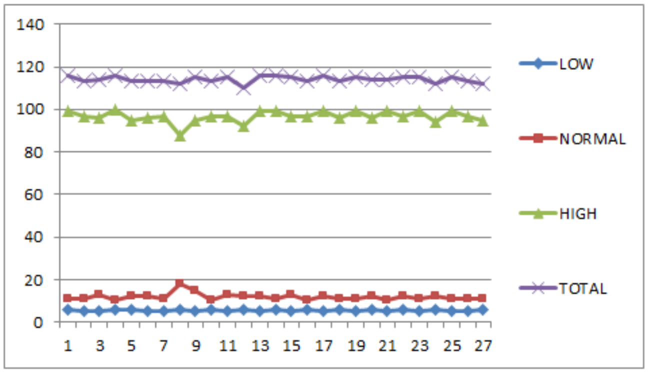

Figure 5. Data transfer division with three requests in a test chunk.

In this chunk of 27 seconds, it is shown that there are three requests running at this moment: a low priority request, a normal priority request and a high priority request. The highest priority request has previously finished so there is 70KB/s free bandwidth available. This free bandwidth is assigned to the high priority request for being the next in the priority list, and the other requests continue with the assigned transfer rate limit. Thus, it can be seen that the request with high priority has a 100KB/s transfer rate.

The results indicate that in most of the cases the prioritization of the network works as intended: the configured minimum transfer rate is complied; ensuring that every request meet the quality of service requirements and when there is free bandwidth it is divided according to priorities level.

Anyway, there are some cases in which the bandwidth is not divided as it has to and it is divided in an equitable way.

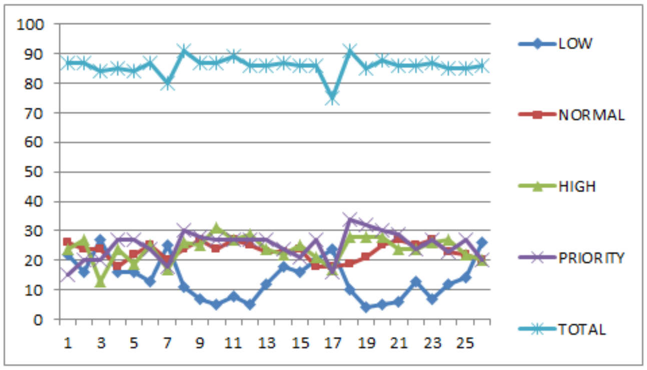

Figure 6. Data transfer division with four requests in a test chunk.

In Figure 6, we can see that the system has 90KB/s of total bandwidth assigned. With four requests of different priorities running in the test scenario, the bandwidth is not divided in a percentage, as it happened in Figure 5, and it is divided in a equitable way. Moreover, it can be seen that the LOW priority request sometimes is limited to its transfer rate, but it is not as constant as it has to be.

This equitable bandwidth division occurs when there is less bandwidth available than the specified one, so the queue disciplines cannot work as they are designed and solve the situation distributing the transfer rate in this way.

According to an intensive analysis of the previously presented test results and the system performance, we can confirm that the proposed communications system has some limitations that must be taken into account in case of a real industrial deployment. For this purpose, the system should be extended adding the abilities described in this section.

Due to the requirements of the chosen queue discipline in the network prioritization, it is necessary to know the bandwidth available at any time. This can be achieved by the method used in this paper, knowing in advance the bandwidth available in each section of the travel path. But this approach only has sense in tests scenarios or in very predictable ones. A more realistic solution could be to have a network monitoring tool that calculates the available bandwidth and updates the network configuration when the bandwidth fluctuates.

It should be noted that this kind of QoS systems (those supported by the set of Linux based software utilities used in this work) were designed to work in static environments, so the development and usage of this monitoring tool is absolutely necessary to adapt it to the current mobile environment. This limitation can be seen in the second result graph (Figure 6) in which the real bandwidth is lower than the specified one, so the queue discipline does not work as it is needed in this vehicular system. GNU/Linux does not have a dynamic QoS system developed [5], so having a network monitoring tool could be a solution to achieve it.

There is another improvement to be considered in this system: the continuous communication. Unlike the common network configurations used in non-mobile environments, such as in an office (where static network links are used, continuous communication can be assured using wired and backup links [2], and the service failure probability depends on rare environmental factors and internet service provider quality), in mobile environments the continuous communication is not as easy to guarantee. The reason is that the communication could be affected by several previously explained dynamic factors.

The best way to assure the network availability is having several 3G modems connected to the system, so if the active link is cut the system can choose another link to continue the communication. In order to accomplish this improvement, the already implemented VPN can be used. The host-to-host VPN tunnel provides an additional abstract layer to the onboard running applications, so after the active link changes the applications will run using the same IP as before the communication link has changed. Furthermore, due to this abstract layer, the link changes would not be detected as a broken link by the onboard applications, thus, the continuous communication between vehicle and ground should be achieved.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the results of two years of research into the design and evaluation of a software system to provide a QoS management solution in dynamic network environments. The tests set-up in real vehicular scenarios, the execution process, the obtained results, and the consequent analysis has been also presented.

The communications system has resulted to be effective in the way in which prioritizes the network traffic, but it has some flaws that have to be fixed to be a fully operational system (in real deployments). Moreover, the lack of ability to ensure continuous communication and to provide an effective active channel change management should be fixed to be a complete vehicular network management system. This evolution of the system should be able to control the QoS of the onboard network and assure continuous communication between vehicles and the traffic control.

Using GNU/Linux utilities, an approximation to the continuous communication challenge could be achieved using the abstraction layer provided by the VPN tunnel, but another network monitoring tool would be needed to change the active link whenever the network coverage is lost.

Related work exists in the area of continuous communication in vehicular environments and also in network request prioritization. This is the case of the software tool [26] developed by our research group, which works as a middleware for monitoring the bandwidth consumption and available mobile network links and subsequently, managing the active channel change.

Nevertheless, the work presented in this paper is also focused on QoS management and prioritization of communications, but reusing already third party developed software and using GNU/Linux as operative system. In conclusion, the objective was to obtain similar results from a new perspective, with GNU/Linux and the tools developed by the open source community in order to reduce programming efforts.

The future work will be focused on two areas. First, on the development of a tool able to dynamically monitor the bandwidth of the network links, adapting the QoS management rules to the bandwidth available in each moment. Second, adding continuous communication abilities to the system, by the adaptation and integration of one of our already existing research projects in this GNU/Linux based system.

Finally, there is also pending work relative to testing the active link change in real scenarios using a VPN as an abstract network layer [27]. It would be the best solution for a GNU/Linux based QoS and network management system as the one proposed here.

ACKNOWLEDGMENT

This work has been funded by the Basque Government of Spain under GAITEK funding program (GEINFEVI project, IG-2011/00472). Special thanks to DATIK - Irizar Group for their support.

REFERENCES

[1] D. Staessens, D. Colle, M. Pickavet, and P. Demeester, “Computation of high availability connections in multidomain IP-over-WDM networks”, ICUMT ’09, International Conference on Ultra Modern Telecommunications & Workshops, Oct. 2009, pp. 1-6.

[2] I. Salaberria, U. Gutiérrez, R. Carballedo, and A. Perallos, “Wireless Communications Architecture for “Train-to-Earth” Communication in the Field of Railways”, DCAI, 2nd International Symposium on Distributed Computing and Artificial Intelligence, Jan. 2009, pp. 625-632.

[3] J. K. -S. Lau, C. -K. Tham, and T. Luo, “Participatory Cyber Physical System in Public Transport Application”, UCC, Fourth IEEE International Conference on Utility and Cloud Computing, Dec. 2011, pp. 355-360.

[4] U. Gutiérrez, I. Salaberria, A. Perallos, and R. Carballedo, “Towards a Broadband Communications Manager to regulate train-to-earth communications”, MELECON, 15th IEEE Mediterranean Electrotechnical Conference, Apr. 2010, pp. 1600-1605.

[5] X. Liu, “Supporting dynamic QoS in Linux”, RTAS, 10th IEEE Real-Time and Embedded Technology and Applications Symposium, May 2004, pp. 246-254.

[6] L. Qi, “Research on Intelligent Transportation System Technologies and Applications”, Workshop on Power Electronics and Intelligent Transportation System, Aug. 2008, pp. 529-531.

[7] F. Benzi, G. S. Buja, and M. Felser, “Communication architectures for electrical drives”, IEEE Transactions on Industrial Informatics, Feb. 2005, vol. 1, pp. 47-53.

[8] M. Felser, “Real-time ethernet - Industry prospective”, Proceedings of the IEEE, 2005, vol. 93, pp. 1118-1129.

[9] R. Ernst, G. Spiegelberg, T. Weber, and H. Kopetz, A. Sangiovanni-Vincentelli, and M. Jersak, “Automotive networks: Are new busses and gateways the answer or just another challenge?”. CODES+ISSS: International Conference on Hardware/Software Codesign and System Synthesis, Salzburg, Sept. 2007, pp. 263.

[10] S. Kurowski, J. Zibuschka, H. Roßnagel, and W. Engelbach, “A Concept for Interoperability of Security Systems in Public Transport”, Proc. of the 9th International ISCRAM Conference, Apr. 2012.

[11] P. Dharwadkar, H. J. Siegel, and E. K. P. Chiong, “A Heuristic for Dynamic Bandwidth Allocation with Preemption and Degradation for Prioritized Requests”, ICDCS, 21st International Conference on Distributed Computing Systems, Apr. 2001, pp. 547-556.

[12] P. Jayachandran and T. Abdelzaher, “Bandwidth Allocation for Elastic Real-Time Flows in Multihop Wireless Networks Based on Network Utility Maximization”, 28th International Conference on Distributed Computing Systems, June 2008, pp. 849-857.

[13] D. Marrero, E. M. Macias, and A. Suarez, “Dynamic Traffic Regulation for WiFi Networks”, Proc. of the World Congress on Engineering, July 2007, pp. 1512-1517.

[14] M.F. Horng, Y.H Kuo, L.C. Huang, and Y.T. Chien, “An Effective Approach to Adaptive Bandwidth Allocation with QoS Enhanced on Ip Networks”, ICUIMC, International Conference on Ubiquitous Information Management and Communication, 2009, pp. 260-264.

[15] P. Noh-sam and L. Gil-Haeng, “A framework for policy-based sla management over wireless LAN”, 2005, Proc. of the Second International Conference on e-Business and Telecommunication Networks, INSTICC Press, ISBN 972-8865-32-5, pp. 173-176.

[16] I. Martínez, “Contribuciones a Modelos de Tráfico y Control de QoS en los Nuevos Servicios Sanitarios Basados en Telemedicina”, Ph.D Thesis, Universidad de Zaragoza, 2006.

[17] I. Salaberria, R. Carballedo, and A. Perallos, "Wireless Technologies in the Railway: Train-to-Earth Wireless Communications", Wireless Communications and Networks - Recent Advances, Ali Eksim (Ed.), DOI: 10.5772/35962, ISBN 978-953-51-0189-5, March 2012, pp. 469-492.

[18] Boss: On Board Wireless Secured Video Surveillance http://celtic-boss.mik.bme.hu/ [retrieved : October 2013]

[19] Ubuntu GNU/Linux: http://www.ubuntu.com/ [retrieved : July, 2013]

[20] OpenVPN, VPN management software: http://www.openvpn.net/ [retrieved : July, 2013]

[21] Ip route, from Iproute2 collection utility: http://www.linuxfoundation.org/ [retrieved : July, 2013]

[22] Netfilter, packet filtering framework: http://www.netfilter.org/

[23] J. L. Valenzuela, A. Monleon, and I. San Esteban, “A hierarchical token bucket algorithm to enhance QoS in IEEE 802.11: proposal, implementation and evaluation”, VTC, IEEE 60th Vehicular Technology Conference, Sept. 2004, vol. 4, pp. 2659-1662.

[24] Webmin: http://www.webmin.com/ [retrieved : July, 2013]

[25] S. S. Kolahi, S. Narayan, D. D. T. Nguyen, and Y. Sunarto, “Performance Monitoring of Various Network Traffic Generators”, UKSim, 13th International Conference on Computer Modelling and Simulation, Apr. 2011, pp. 501-506.

[26] I. Salaberria, A. Perallos, and R. Carballedo, “Towards a Dynamic and Adaptative Prioritization of Wireless Broadband Vehicle-to-Ground Communications”, ACCESS, The 3th International Conference on Access Networks, June 2012, pp. 31-34.

[27] G. Urquiola, A. Perallos, and R. Carballedo, “Continuous Broadband Communication System Base on Existing Open Source Network Tools for Vehicular Environments”, ITSC, 15th International IEEE Conference on Intelligent Transportation Systems, Sept. 2012, pp. 248-253.