On the Development of Dependable Embedded Applications Using Specialized Wormholes

António Casimiro casim@di.fc.ul.pt FC/UL
Odorico M. Mendizabal omendizabal@di.fc.ul.pt FC/UL
Paulo Veríssimo pjv@di.fc.ul.pt FC/UL*

Abstract

The development of embedded technologies to support the construction of dependable applications in environments of uncertain synchrony, reliability or security, raises many challenges. In previous work we introduced a framework based on the concept of wormholes, and some techniques for adaptation and fail-safety to construct dependable applications under this framework. But in order to set up a wormhole, it is necessary to enforce a hybrid system structure through the use of appropriate hardware and engineering approaches. This paper gives some initial steps towards the definition of new wormholes specifically suited for embedded systems. We first describe the framework that we intend to follow and then we discuss some existing processing and communication technologies that could serve our purposes.

1. Introduction

The development of embedded technologies to support cooperative, ubiquitous, pervasive and yet dependable applications, raises many technical and scientific challenges. For instance, when dealing with uncertainty of the environment, it is hard to provide any guarantees, especially non-functional, such as performance, timeliness, reliability or security. This is particularly true of environments of uncertain topology, timeliness or hostility, such as those found in intelligent transportation systems, involving car-to-car communication (C2CC) or car-to-infrastructure communication (C2IC).

Current state-of-the-art uses classical fault tolerance for dependability, and QoS management as the workhorse for adaptation. In fault tolerance, the approach is normally based on fairly static assumptions on system structure and fault assumptions. In QoS, the approach is normally heuristic, based on aprioristic assumptions on the system and its environment, and done at a partial level.

Architectural frameworks are needed that handle uncertainty and foster the concept of adapting functional and non-functional properties of a service while at least providing guarantees on how dependably it is adapting, and of doing it on an end-to-end, systemic perspective, rather than e.g., network or host based. For example, from the perspective of timeliness, this can be addressed by a suite of mechanisms and protocols based on fail-safety, time-safety and time-elasticity, to perform fault tolerance and dependable QoS assurance and adaptation in settings where the need for some real-time behavior conflicts with the uncertainty about the environment's timeliness. However, exactly the same can be said of security and fault tolerance vs. malicious faults, or better put, intrusion tolerance.

The state-of-the-art should be advanced with system models for embedded and mission critical operation, which provide the means to deal with the uncertainty of the environment, while addressing real-time and consistency needs of applications. To this end, in previous work we introduced *wormholes* [18] as enhanced subsystems which provide components with a means to obtain a few simple privileged functions and channels to other components, with "good" properties otherwise not guaranteed by the "normal" weak/uncertain environment. For example, providing timely or secure functions and communication in, respectively, asynchronous systems or byzantine-on-failure environments.

The wormhole concept can be exploited for the construction of applications with timeliness or security requirements, through architectural hybridization, a well-founded way to substantiate the provision of those "good" properties on "weak" environments. In fact, specific instantiations of wormholes have been presented and discussed in earlier work [4, 6].

Now we want to further exploit the concept, by demonstrating the feasibility of other wormholes and their suitability to support the development of dependable embedded

*Faculdade de Ciências da Universidade de Lisboa. Bloco C6, Campo Grande, 1749-016 Lisboa, Portugal. Navigators Home Page: http://www.navigators.di.fc.ul.pt. This work was partially supported by the EC, through the HIDENETS project, IST-FP6-STREP-26979.

applications involving ad-hoc C2CC with infrastructure service support. This paper presents some initial steps in this direction, namely with a brief description of the approach that will be followed to deal with uncertainty through dependable adaptation, and with an overview of COTS-based technologies that may be used to construct the necessary wormhole.

The reminder of the paper is organized as follow: Section 2 introduces hybrid distributed system models based on wormholes. Section 3 discusses dependable QoS adaptation and how it uses wormholes. In Section 4 we analyze COTS-based communication and embedded technologies from an architectural hybridization perspective. Finally, Section 5 provides some final remarks and next steps.

2. Hybrid Distributed System Models

Since we are considering that the environment surrounding distributed applications is open, unpredictable or uncertain, classical distributed system models are typically not able to reconcile this uncertainty with the needs for predictability observed in many applications. For example, when reasoning in terms of synchrony requirements, if a fully synchronous model is used then the application will fail as soon as an assumed bound is violated (and assuming very high bounds to reduce the probability of failure can make the application too slow and unusable). On the other hand, fully asynchronous models do not even allow to deal with timeliness requirements, while making it impossible to solve relevant problems such as consensus in the presence of a single crash failure [10].

A promising way to address this conflict between the need for predictability and the uncertainty of the environment consists in considering models of partial synchrony or, in a more generic way, hybrid distributed system models. An hybrid model assumes that the system is not homogeneous – its properties, like synchrony or trustworthiness, can vary over time or with the part of the system being considered. In fact, the concept of wormholes [18] has been introduced to explain how this architectural hybridization can be used to find better solutions for fundamental distributed system problems and to improve the construction of dependable applications.

When applying the concept of wormholes to the synchrony domain, a system is assumed to have distinct parts with respect to synchrony properties. There should be a synchronous, or at least timelier part, very small and using just the needed resources to provide a few crucial services, which constitutes the wormhole (we also call it the *control* part). When considering distributed wormholes, synchrony properties must be secured in a distributed way, which requires also some form of hybridization of the communication infrastructure. General applications are executed on the other part of the system, a possibly complex and asynchronous part (to which we call *payload*), making use of the services provided by the wormhole (through a local instantiation, or interface) only when this is needed to secure some critical property. Naturally, this implies the definition of specific interfaces and an adequate programming model to use these wormhole services. Possible services include the measurement of distributed durations, the detection of timing failures or even the execution of specialized functions in a timed and/or timely way.

In order to use the wormhole model and exploit architectural hybridization, it is necessary to ensure, by design, that the assumed properties for the wormhole are satisfied. In practice, hybridization must be enforced either by identifying and isolating the parts of the system that may constitute the wormhole, or by adding the necessary amount of resources, be it processing, communication or other, that are needed to construct the wormhole.

There is certainly a vast range of solutions for achieving the required improved subsystems. For example, the availability of a GPS-based clock can be used as an alternative time base to the local system clock. Improved communication properties for the wormhole can be secured by the use of additional dedicated communication channels or networks or simply using networks with differentiated services. Dedicated processor boards or devices with embedded processors can be used to implement local instances of a wormhole, as an alternative to sharing the local processor between the wormhole and the rest of the system, as it could be done with appropriate OS support. Depending on the kind of application and on the environment in which it will execute, different wormholes can be envisaged. We will be further discuss this issue in Section 4, considering some possibilities available for the construction of wormholes for embedded systems.

3. Dependable QoS Adaptation

Classical real-time approaches for QoS provision assume that resource reservation is possible. This is acceptable in environments in which it is possible to control available resources. However, since we assume dynamic and essentially unpredictable environments, it is difficult to secure time bounds because, in fact, no guarantees can be given that the necessary amount of resources is available all the time. Therefore, unlike the majority of the work dealing with QoS provision, we follow a different approach, assuming that the best that can be done is to adapt applications according to the amount of resources that are available at a certain moment or that were available during some interval of observation. In that sense, we do not try to do things such as rearranging or redistributing the existing resources as a means to try to secure a given QoS for the application.

Instead, we rely on the application to adapt itself, if possible, to the actual conditions of the environment.

In fact, we also consider different approaches for dependable application programming in these uncertain environments. For instance, we also consider a fail-safety approach, in which applications may not be able to adapt but may be able to stop in a fail-safe state when requirements cannot be satisfied, or a redundancy-base approach, in which relevant timed actions are replicated to the necessary amount (given the observed available resources) for securing certain timeliness requirements.

Here, however, we only focus on adaptable applications. We call them *time-elastic*, to express the idea that they are able to dynamically increase or decrease the assumed time bounds. The success of our approach for adaptation has to do essentially with two factors: (i) the monitoring framework, which dictates the accuracy of the observations leading to adaptation and (ii) the adaptation framework, which determines how and when should bounds be adapted. While the later is fundamentally based on probabilistic, statistical and mathematical solutions to compute adaptation parameters, the former strongly depends on the system architecture and on the supporting infrastructure (e.g., the network and the operating system). This is why we consider an architecture based on the existence of an wormhole, and we discuss in this paper the possible solutions for implementing it.

In [5] a framework for dependable QoS adaptation using a specific timeliness wormhole, a TCB (Timely Computing Base), was presented. This wormhole can be used essentially to help the construction of the monitoring framework, by providing the adequate means to measure distributed durations with a given accuracy, and independently from the "observed" system. This provides the basis to achieve what we call *dependable adaptation*, that is, the ability to provide some guarantees on how adaptation should be done.

The adaptation framework is based on the construction of probability distribution functions (*pdf*) that represent the actual distribution of the observed timing variables, which are then used to determine the specific probability of ensuring that a given timing bound can be secured. Adaptation is then carried on with the objective of keeping this probability always constant during the overall execution of an application. In other words, the goal is to ensure *coverage stability*, and in this way allow the development of dependable embedded (adaptive) applications in the assumed uncertain environments [19].

4. Technologies for Embedded-System Wormholes

In this section we discuss suitable technologies to develop embedded systems based on the wormhole model. In particular, we are interested in COTS-based embedded devices for communication, timing and processing support. In order to motivate our analysis, we consider a scenario involving car-related applications and including Car-to-Car Communications (C2CC) and Car-to-Infrastructure Communications (C2IC).

A broad range of inter-vehicle applications could be considered. Hazard warnings, collision avoidance and intelligent traffic signs are examples of applications with time- and security- dependent safety requirements. Interactive multimedia applications can also be envisaged due to their timeliness requirements. In order to design these applications under the wormhole framework, the application code and the application specific communication workload has to reside on the payload part of the system, while the control part will be responsible for performing important actions for the safety of the system, like measuring delays or detecting timing failures. In general, a wormhole would be used for monitoring and adaptation of C2CC, as explained in Section 3. But it could also be used for in-car environments, improving the safety of operations involving distributed and accurate sensor readings or timely actuation. This would require the use of a timely timing failure detection service.

Figure 1(a) depicts a car equipped with several real-time sensor networks. Black rectangles represent actuators and sensors, connected through a network. Clearly, the synchrony of every network should match the timeliness bounds required by each application. For instance, the ABS and gearbox actuators must be more accurate and timely than door locks. While it is possible to use classic real-time approaches to implement these applications in such a controlled environment, if some of this information has to be disseminated to other cars nearby, or to the infrastructure, then the availability of a wormhole will help the application designer.

Figure 1(b) depicts the whole system. Now the hybrid nature of the systems is revealed through the existence of payload and control parts represented in every car and in some equipment on the same environment. Inside the car there must exist a wormhole instance acting as a gateway between the several internal car networks and the outside environment. On this outside environment, messages can be exchanged by payload parts between cars, or between cars and other equipment. The control parts must exchange information through own communication channels, isolated from the payload. The existence of specialized devices (e.g. GPS, as represented in the picture) may be made part of the control subsystem. The picture also represents the interface between payload and control, since the problem of interfacing a potentially asynchronous environment to a more synchronous one is not trivial.

Nowadays mobile network technologies present different capabilities and can be suitable to devise systems based in wormholes. WLAN (e.g. IEEE 802.11 [7, 12] and Blue-

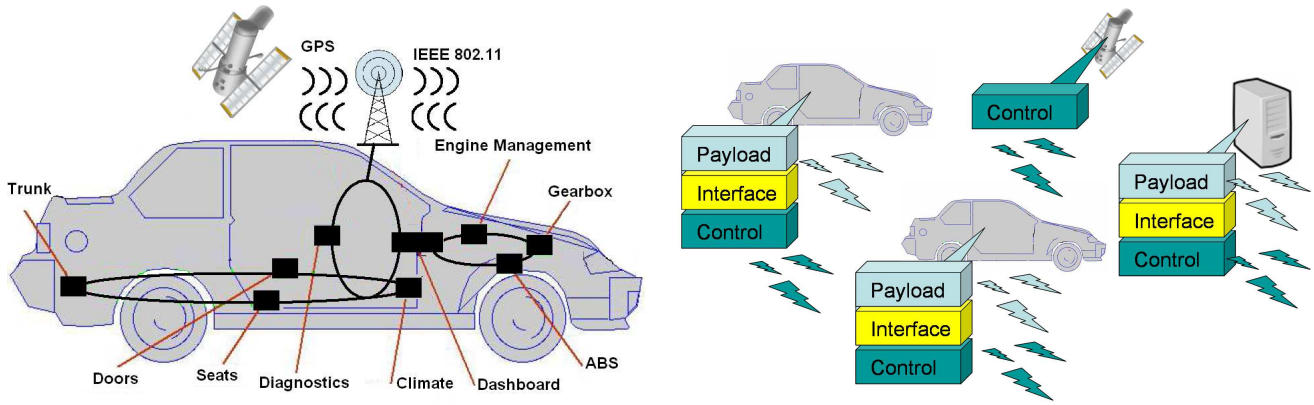

Figure 1. Application scenario: (a) car networks; (b) hybrid architecture.

tooth [11]) or cellular networks (e.g. UMTS [2, 8, 15]) are some examples of wireless networks that can be integrated in order to achieve different synchrony or security levels. Although WLAN and cellular networks present distinctions regarding technology, frequency range, cost, among other parameters, we are particularly interested in their coverage range and transfer rate. The former present a limited range, but higher transfer rate. On the contrary, cellular networks have a higher coverage but reduced transfer rate [8].

In order to improve the availability of server-based applications and, in general, to secure the better possible QoS for the application, it is possible to consider the existence of multiple communication channels serving the payload part. In such setting, the availability of a dependable adaptation framework could be used to assist reconfiguration of the communication subsystem, aiming at using the channel providing the better QoS.

IEEE 802.11 standards [7, 12] are envisaged here as suitable for ad hoc C2CC or even C2IC. Of particular interest is the IEEE 802.11p (standard for Wireless Access in Vehicular Environments) [1], which seems well suited for our purposes.

To enforce better properties on the communication channels, some features of IEEE 802.11p are handy, (e.g. different priorities levels, multi-channel operations and applications differentiation). For instance, IEEE 802.11p transmission range is divided into 7 channels of 10 MHz each, with 1 dedicated control channel and 6 service channels. In addition, it is possible to combine two channels to increase the frequency band.

Cellular network channels are suitable to disseminate information in a broad range. While they may not be very suited to implement timely or synchronous channels (in particular, because they can be affected by interference and disconnection quite easily), we believe they may be nevertheless used to better secure trustworthiness requirements of payload applications, through a specific wormhole for this purpose. For instance, secure cryptographic keys could be refreshed periodically based in secure, deterministic and wide area communication established between cars and infrastructure (e.g. VPN constructed over UMTS technologies). Solutions found in the literature addressing resilient approaches based on hybrid distributed system models should also be investigated in wireless scenarios. This is the case of proactive recovery [17] and strategies to solve consensus [16].

Considering wireless communication in a short range or even inside a car, Bluetooth wireless technology [11] is well suited to build payload or control channels. To achieve the synchronism necessary for the control part, synchrony links (SCO) should be obtained using Bluetooth devices sharing a common channel [3]. Inside the cars other solutions can also envisaged, namely using hybridization of a field-bus network, such as FTT-CAN [9]. FTT-CAN combines both time- and event-triggered paradigms, taking advantage of the underlying medium-access control of CAN to implement channels of differentiated guarantees.

With respect to timing, specific solutions must also be investigated. The use of GPS, as already mentioned, is a good solution to establish a global and accurate time base. On the other hand, assuming the availability of GPS receivers is reasonable since we consider a scenario involving car related applications. Specific solutions for timing, possibly based on software clock synchronization, or the availability of other time provision networks, must be sorted out when considering closed spaces, such as tunnels or underground parking lots.

An earlier solution to construct a system with a control and a payload processing environment was presented in [4]. There, a TCB implementation using RT-Linux was

described, focusing on the specific problems of enforcing the required properties for the control part. But a wormhole can in fact be implemented in other ways, not using a software-based technique to create virtually separated environments, but using real separate execution environments, with their own processors. For instance, PC hardware can use co-processors, being combined with embedded systems or micro-controllers to create hybrid systems with regard to processing. The computational power of today's embedded co-processors motivates the development of hardware-based hybrid systems.

There exist many embedded and versatile boards (for example, the Kontron ePCI 101 [13] presents a dual Ethernet interface, 1.6 GHz CPU clock and 1 GB DDR-SDRAM) that could be added to a PC system in order to address our objectives. Other PC interfaces can also be used to connect even more independent micro-controllers or embedded systems. For instance, the Lantronix UDS100 [14] is a small embedded device with reasonable processing capabilities, which can be connected to a PC through a serial interface, and may be linked to similar devices through a dedicated ethernet network, thus forming a form of embedded wormhole. With appropriate bridging, it would also be possible to envisage a system made of several UDS100 devices connected through a wireless network. This could possibly be a good solutions for the applications and environments we consider in this paper.

5. Concluding Remarks

In this paper we motivated the use of the wormhole distributed system model and framework to implement embedded applications in environments of uncertain timeliness. To this end, we also revisited the use of dependable QoS adaptation as an appropriate solution for some applications running over such unpredictable environments. Scenarios involving car related applications were considered to motivate a discussion about possible COTS-based solutions to implement a wormhole based system.

We are currently in the process of implementing a new wormhole better suited to embedded systems and applications, and we plan to use and demonstrate these results in the scope of the HIDENETS [20] European project. Along with this work, we also plan to further improve previous results on dependable adaptation, in particular be investigating suitable probabilistic distributions for particular car-to-car environments and for the specific envisaged applications.

References

[1] IEEE Standard 802.11p Draft Amendment. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Wireless Access in Vehicular Environments (WAVE), 2005.

[2] R. Berezdivin, R. Breinig, and R. Topp. Next-generation wireless communications concepts and technologies. *Communications Magazine, IEEE*, 40(3):108–116, 2002.

[3] P. Bhagwat. Bluetooth: Technology for short-range wireless apps. *IEEE Internet Computing*, 5(3):96, 2001.

[4] A. Casimiro, P. Martins, and P. Veríssimo. How to build a timely computing base using real-time linux. IEEE International Workshop on Factory Communication Systems, 2000.

[5] A. Casimiro and P. Veríssimo. Using the timely computing base for dependable qos adaptation. In *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, pages 208–217, 2001.

[6] M. Correia, P. Veríssimo, and N. F. Neves. The design of a COTS real-time distributed security kernel. In *Proceedings of the Fourth European Dependable Computing Conference*, pages 234–252, 2002.

[7] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. IEEE 802.11: Wireless local area networks. *IEEE Communications Magazine*, 35(9):116–126, 09 1997.

[8] J. De Vriendt, P. Laine, C. Lerouge, and Xiaofeng Xu. Mobile network evolution: a revolution on the move. *Communications Magazine, IEEE*, 40(4):104–111, 2002.

[9] J. Ferreira, P. Pedreiras, L. Almeida, and J. A. Fonseca. The FTT-CAN protocol for flexibility in safety-critical systems. *IEEE Micro*, 22(4):46–55, 2002.

[10] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

[11] J. C. Haartsen. The bluetooth radio system. *Personal Communications, IEEE*, 7(1):28–36, 2000.

[12] IEEE Standards Association. IEEE 802.11 wireless local area networks. http://grouper.ieee.org/groups/802/11/, checked 2006-07-20.

[13] Kontron. ePCI 101. http://emea.kontron.com/index.php?id=226&cat=48&productid=207, checked 2006-07-20.

[14] LANTRONIX. UDS 100. http://www.lantronix.com/device-networking/external-device-servers/uds-10.html.

[15] H. Luediger and S. Zeisberg. User and business perspectives on an open mobile access standard. *Communications Magazine, IEEE*, 38(9):160–163, 2000.

[16] N. F. Neves, M. Correia, and P. Veríssimo. Solving vector consensus with a wormhole. *IEEE Transactions on Parallel and Distributed Systems*, 16(12):1120–1131, December 2005.

[17] P. Sousa, N. F. Neves, and P. Veríssimo. Proactive resilience through architectural hybridization. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC)*, pages 686–690, 2006.

[18] P. Veríssimo. Travelling through wormholes: a new look at distributed systems models. *SIGACT News*, 37(1):66–81, 2006. http://www.navigators.di.fc.ul.pt/docs/abstracts/verissimo06travelling.html, checked 2006-07-20.

[19] Paulo Veríssimo and António Casimiro. The timely computing base model and architecture. *IEEE Trans. Comput.*, 51(8):916–930, 2002.

[20] HIDENETS web page. http://www.hidenets.aau.dk/, checked 2006-07-20.