A Model-Based Approach to the Autonomic Management of Mobile Robot Resources

Adolfo Hernando, Ricardo Sanz
Autonomous Systems Laboratory
Universidad Politécnica de Madrid
José Gutiérrez Abascal 2
Madrid 28006, Spain
Adolfo.Hernando@upm.es, Ricardo.Sanz@upm.es

Radu Calinescu
Aston University
Aston Triangle
Birmingham B4 7ET, UK
R.C.Calinescu@aston.ac.uk

***Abstract*—Mobile robots are increasingly used in applications as diverse as space exploration, rescue missions, automatic floor cleaning in buildings and factories, and mobile surveillance systems. However, one major challenge encountered in the development of applications involving mobile robots is the limited amount of resources that these devices have at their disposal. Effective use of mobile robot resources such as bandwidth is essential for the success of these applications. This paper presents a model-based, utility-driven approach to the autonomic management of mobile robot resources. Starting from a model of the robot resources and of the components that use these resources, the approach builds an autonomic solution capable of optimising a utility function supplied by the robot administrator. This self-optimisation involves adapting the operation mode of the robot components to changes in the amount of resources available and in the environment. The approach is validated through a case study that involves the self-optimising management of bandwidth for a Pioneer 2AT-8 mobile robot.**

***Keywords*-model-based self-X adaptation, mobile robot, utility function, resource management**

I. INTRODUCTION

As mobile robots are becoming increasingly sophisticated, they acquire the potential to be used in a growing number of applications. For instance, over the past few years our research group at the Universidad Politécnica de Madrid has developed software for the remote control of a Pioneer 2AT-8 robot [1], [2] and its on-board devices. Equipped with this software, the mobile robot offers an increased number of possible configurations. Each of these configurations corresponds to certain on-board devices being active, and is appropriate for different scenarios that the robot may operate in. Furthermore, selecting one or another of these configurations has a different impact on the resource utilisation for the robot as a whole. Some configurations may reduce the time interval that the robot may operate without recharging its batteries, while others may be impossible unless the wireless network used to communicate between the robot and the fixed workstations in the laboratory operate at full capacity.

Taking advantage of these new robot capabilities requires the use of configurations that are dependent on the current state of the robot and its environment, as well as on the objectives of the application that it is involved in. This is a problem widely faced by developers of mobile-robot applications [3], [4].

This paper focuses on the use of autonomic, self-optimising management of robot resources to address the problem described above. An autonomic solution was considered ideal for managing robot resources due to its potential to relieve the human operator from the burden of manually and repeatedly reconfiguring the on-board robot devices. Instead, the operator would be able to specify high-level objectives to be implemented by an *autonomic manager* through monitoring the state of the system, analysing the current state, planning changes required to achieve these objectives and executing these changes. This so-called *monitor-analyse-plan-execute* or MAPE autonomic computing loop is described in detail in [5], [6].

The model-based approach to autonomic resource management introduced in this paper uses the GPAC (General-Purpose Autonomic Computing) platform for the development of autonomic computing applications [7] and defines the objectives for the autonomic management of the robot resources in terms of *utility functions* [8]. The main contributions of the paper include integrating the mobile robot with the GPAC platform, building a model for the self-optimising bandwidth management for a mobile robot, and using it to develop an autonomic resource-management solution based on a set of utility-function autonomic policies. Extensive experiments were carried out to evaluate the effectiveness and overheads of this solution.

The remainder of the paper is organised as follows. Section II describes the mobile robot used in this work, and the GPAC autonomic computing platform. Related work is then presented in Section III, followed by a description of our approach and of the experimental results in Sections IV and V, respectively. Section VI concludes the paper with a brief summary and a discussion of future work.

II. BACKGROUND

A. The Pioneer 2AT mobile robot

The work described in this paper was carried out using the Pioneer 2AT-8 mobile robot shown in Figure 1, and

 ISBN: 978-1-61208-109-0

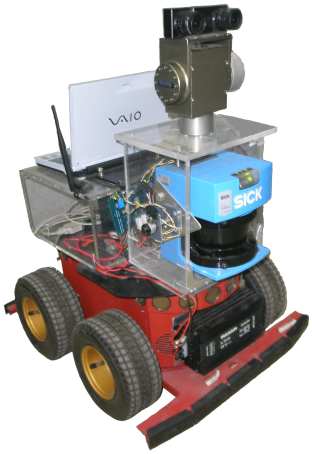

Figure 1. The Pioneer P2AT-8 robot

manufactured by ActivMedia Robotics [9].

The Pioneer 2AT-8 is a skid-steering vehicle with lightweight aluminium body and four pneumatic wheels powered by four reversible DC motors, equipped with high-resolution optical encoders for precise positioning, speed sensing and advanced dead-reckoning. The on-board electronic devices are controlled by a Hitachi H8S microcontroller.

Additionally, the Pioneer 2AT-8 robot has two sonar arrays that provide object detection and range information for collision avoidance. There are two computers on-board the mobile robot, and the experimental setting includes several workstations connected to a local area network and used to control the robot remotely. The on-board computers and the workstations communicate using distributed objects enabled by the Common Object Request Broker Architecture (CORBA) Object-Oriented Middleware, and more specifically the Object Management Group's CORBA 3.1 specification [10].

The control system on the mobile robot enables fine-grained control of the on-board devices through a flexible API implemented by our research group, and which can be accessed via the two computers. This control includes the ability to switch on and off components such as the video camera and the laser SICK[1] or to adjust configuration parameters such as the frames-per-second parameter for the video device. The project described in this paper uses this functionality to adapt the configuration of the robot to changes in the bandwidth of the wireless link between the robot and the workstations used to control it remotely.

The self-adaptation of the robot configuration described above is carried out such as to maximise an analytically defined robot *utility*, subject to not exceeding the available bandwidth. The specification of a utility function for this purpose, and the techniques used to add self-adaptation capabilities to the Pioneer 2AT-8 mobile robot are described in Sections IV and V, respectively.

B. Autonomic application development with GPAC

GPAC is a service-oriented software platform for the model-driven development of self-* systems [12]. GPAC has been used to develop self-* systems in application domains ranging from autonomic data-centre infrastructure management and dynamic power management to data-storage allocation for military vehicles [13], [14], [15].

The core component of the platform is a reconfigurable autonomic manager capable of augmenting existing IT systems with a MAPE autonomic computing loop. This is achieved using automated code generation techniques to synthesise new software components starting from a model of the IT system that is supplied to the autonomic manager as part of a run-time configuration step [16]. *GPAC system models* describe formally (a) the characteristics of every relevant parameter of the system, including its name, type (i.e., to be `monitored` or `configured` by the MAPE loop) and value domain (e.g., `integer`, `double` or `string`); and (b) the run-time behaviour of the system.

The GPAC autonomic manager supports several types of autonomic computing policies, including action policies, goal policies and utility-function policies [12]. The project described in this paper uses GPAC utility-function policies. To illustrate the specification of this type of autonomic computing policy,[2] and without loss of generality, we will consider policies requiring the optimisation of *multi-objective utility functions* of the form:

$$utility = \sum_{i=1}^{n} w_i \, objective_i, \qquad (1)$$

where the *weights* $w_i \geq 0$, $1 \leq i \leq n$, are used to express the trade-off among the $n > 0$ system objectives. Each objective function $objective_i$, $1 \leq i \leq n$, is an analytical expression of (configurable and monitored-only) system parameters defined in the GPAC system model.

III. Related work

Utility functions for autonomic systems were originally introduced in [8], and have since been used to build self-managing resource allocation solutions by a number of projects [17], [18]. However, none of the projects that we are aware of has explored the use of utility-based adaptation in managing mobile-robot resources.

[1] A SICK Laser Measurement Sensor (LMS) 200 is an accurate distance measurement sensor that is quickly becoming a staple of the robotics community [11]. The LMS 200 can easily be interfaced through RS-232 or RS-422, providing distance measurements over a 180 degree area up to 80 meters away. The sensor operates by shining a laser off of a rotating mirror. As the mirror spins, the laser scans 180°, effectively creating a fan of laser light.

[2] See [12] for details about the other policy types supported by the GPAC autonomic manager.

 ISBN: 978-1-61208-109-0

A bandwidth-allocation approach that employs price models to optimise the cost of bandwidth usage is proposed in [19]. This optimization is performed on a network in which there are data packets from different devices that may use different routes. Our problem is different, as it involves a unique route: the wireless network has one access point and a single, multi-device client (the robot). Therefore, our solution optimises the bandwidth usage on this network for the data streams originating from the different, competing devices of the mobile robot.

Other research work addresses the problems of routing in ad-hoc networks (e.g., [20]). These problems are not analyzed in this paper, since we use a managed wireless network, without routing during transmission / reception of wireless data.

IV. Approach

In our laboratory, the mobile robot was provided with an electronic board that allows to switch on and off the on-board laser range finder, the robotic wrist, the video camera, and the GPS devices programmatically. Note that each of these devices is required only when the robot performs certain tasks, and that they each utilise a specific amount of the bandwidth available between the on-board computers and the fixed control workstations in the laboratory.

A. Introduction

Our aim was to build a self-managing bandwidth allocation solution enabling the robot to perform its tasks optimally when the available bandwidth is insufficient for the correct operation of all on-board devices. To achieve this, we built a model of the robot that describes the parameters relevant for our application:

- $video \in \{0, 1\}$—a configuration parameter that is 1 if the video camera is switched on, and 0 otherwise;
- $fps \in [0, 200]$—the video frames-per-second rate;
- $width \in \{640, 1280\}$—the video width;
- $rgb \in \{0, 1\}$—the video color space, which is 0 if the color space is Bayer, and 1 if the color space is RGB;
- $laser \in \{0, 1\}$—a configuration parameter that is 1 if the laser SICK is switched on, and 0 otherwise;
- $sonar \in \{0, 1\}$—a configuration parameter that is 1 if the sonar arrays are switched on, and 0 otherwise;
- $brate \in [0, 300]$—a state parameter that specifies the available bandwidth, in Mbits/s;

This model (Figure 2) was used to configure an instance of the GPAC autonomic manager running on one of the workstations in the laboratory. A *manageability adaptor* was implemented using a combination of C# and C++ to allow the autonomic manager to monitor and configure the on-board robot devices.[3]

[3]GPAC manageability adaptors are thin, web-service interfaces that implement a small set of predefined monitoring and configuration operations using the native API of the managed component(s) within a GPAC autonomic system [16].

```
<?xml version="1.0" encoding="UTF-8"?>
<system xmlns="http://www.rcc.com/system"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.rcc.com/system file:/C:
    <name>BandwithControl</name>
    <resource>
        <ID>robot</ID>
        <!-- state parameter section -->
...
        <property>
            <ID>fps</ID>
            <propertyDataType>
                <xs:element name="fps" type="robotfps" />
                <xs:simpleType name="robotfps">
                    <xs:restriction base="xs:double"/>
                </xs:simpleType>
            </propertyDataType>
            <mutability>constant</mutability>
            <modifiability>read-only</modifiability>
            <subscribeability>false</subscribeability>
            <primaryKey>false</primaryKey>
        </property>
        <!-- derived parameter section -->
        <property>
        <!-- configuration parameter section -->
        <property>
            <ID>rgb</ID>
            <propertyDataType>
                <xs:element name="rgb" type="robotrgb" />
                <xs:simpleType name="robotrgb">
                    <xs:restriction base="xs:int"/>
                </xs:simpleType>
            </propertyDataType>
            <mutability>constant</mutability>
            <modifiability>read-write</modifiability>
            <subscribeability>false</subscribeability>
            <primaryKey>false</primaryKey>
        </property>
...
        <!-- operational model section -->
        <property>
    </resource>
</system>
```

Figure 2. GPAC mobile robot model.

In order to use GPAC utility-function policies as described in Section II-B, we also generated an *operational model* describing the relationship between the on-board devices that are switched on and the amount of bandwidth that these devices require. GPAC supports the specification of operational models either as tables whose entries reflect this relationship between parameter values or as analytical Markovian models. This application used the former type of operational model, which was built starting from the equations that relate the bandwidth usage to the configurations relevant for the scenarios in which the mobile robot was envisaged to operate.

B. Utility functions

Two variants of the utility function in eq. (1) were used. We initially used a simple utility function defined as a linear combination of the video, sonar and laser parameters of the robot, and assuming a fixed value for the 'video frames per second' parameter (i.e., $fps = 3$):

$$utility_1 = (w_1 width + w_2 rgb) video + w_3 sonar + w_4 laser \quad (2)$$

subject to the overall bandwidth usage not exceeding the available bandwidth $brate$:

$$\frac{1024^2}{8} brate \geq video(rgb\ ?\ 3 : 1) width^2 0.75 fps + \\ + 1280\ sonar + 217200\ laser$$

where

$$\begin{aligned} 1280 =& 16 \text{ sonars} * 4 \text{ bytes/sonar} * 20 \text{ Hz} \\ 217200 =& 181 \text{ measures} * 8 \text{ bytes/data} \\ & * 2 \text{ data/measure} * 75 \text{ Hz} \end{aligned} \quad (3)$$

and the notation $x?y : z$ denotes a function that takes value y if its boolean argument x is true and z otherwise.

The weights w_i, $1 \leq i \leq 4$, depend on the scenario where the robot is present. This weights are determined by considering the data transmitted over the network link. During daytime, w_1 and w_2 are positive constants, appropriate for light conditions. In contrast, when the robot operates in a dark environment (e.g., during the night), the weights are set to $w_1 = w_2 = 0$ because using the camera does not contribute to the utility of the robot.

We also experimented with utility functions that take into account the fps frames per second parameter and use exponentials to express the utility of the video camera more accurately:

$$utility_2 = [w_1 A_1(fps) + w_2 A_2(fps) width + \\ + w_3 A_3(fps) rgb] video + w_4 sonar + w_5 laser$$

where

$$A_i(fps) = \left[\left(1 - \frac{1}{1 + \exp\left(K_i * (fps - L_i)\right)} \right) - M_i \right]$$

and $K_i, L_i, M_i > 0$ for all $1 \leq i \leq 3$. (4)

This utility functions (Figure 3) is a significant improvement over $utility_1$ because it expresses the fact that very low frames-per-second video has little utility, while the added value of increasing fps grows quickly in the middle of the acceptable range for this parameter. Furthermore, $utility_2$ encodes the fact that increasing fps above a certain value is of limited utility, as the extra amount of video data acquired cannot be processed in real time using the available CPU and memory resources.

Note also that, due to its sigmoid-like shape, $utility_2$ has the additional advantage that it does not suffer from stability problems when the self-optimising system decides the value for the fps parameter. Adding an $w_5 fps$ term to the simpler utility function $utility_1$ was found to suffer severely from such stability problems.

V. Experimental results

We simulated the network conditions of the robot using real data from the wireless network adapter of a laptop computer similar to the computers on board the mobile robot. Figure 4 depicts three typical experiments using different such data sets for the bandwidth bit-rate and the two utility functions from Section IV-B. This experiments have been chosen to represent, in three small experiments, a broad spectrum of all possible values of the set of available bandwidth values and its variation over time. The first row of diagrams in this figure shows the available bandwidth for the three experiments. The results when $utility_1$ and $utility_2$ were used are illustrated by the diagrams in the second and third row in Figure 4, respectively. Additionally, Tables I and II summarise the results of a larger number of experiments. Note that because the robot utility function can be modified by its administrator at any time, it is not possible to pre-compute the optimal configurations.

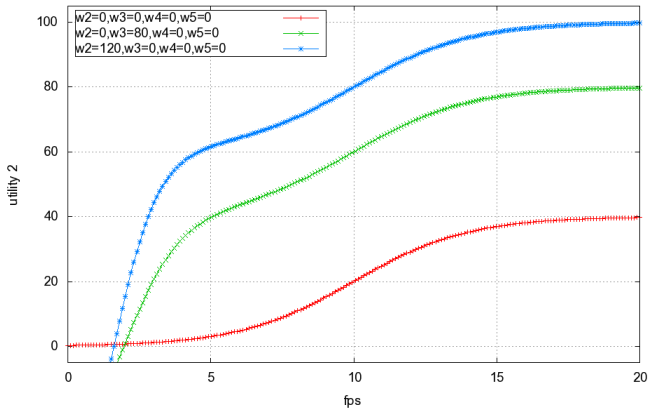

Figure 3. The shape of $utility_2$ as a function of the fps parameter (the weights and the other parameters are fixed: $w_1 = 40$, $K_1 = 0.5$, $L_1 = 10$, $M_1 = 0$ $K_2 = 1.3$, $L_2 = 5$, $M_2 = 0.5$ and $K_3 = 1.05$, $L_3 = 2$, $M_3 = 0.5$.)

A. Discussion

The results obtained using the two utility-function policies are apparently very similar across the three experiments. When extremely low bandwidth is available (1Mbps), only the sonars are operational, while an increased amount of available bandwidth results in the laser and then the video being switched on. However, there is also a significant difference between the two policies. When $utility_2$ is used, the video is always switched on, and the frames-per-second fps parameter is adjusted to ensure that the mobile robot copes with the available bandwidth. In contrast, recall that $utility_1$ uses a fixed value of $fps = 3$. For low available bandwidth, this fps value is too large, and the video is switched off, bringing the robot into an operating mode in which the range of tasks it can execute is significantly reduced.

 ISBN: 978-1-61208-109-0

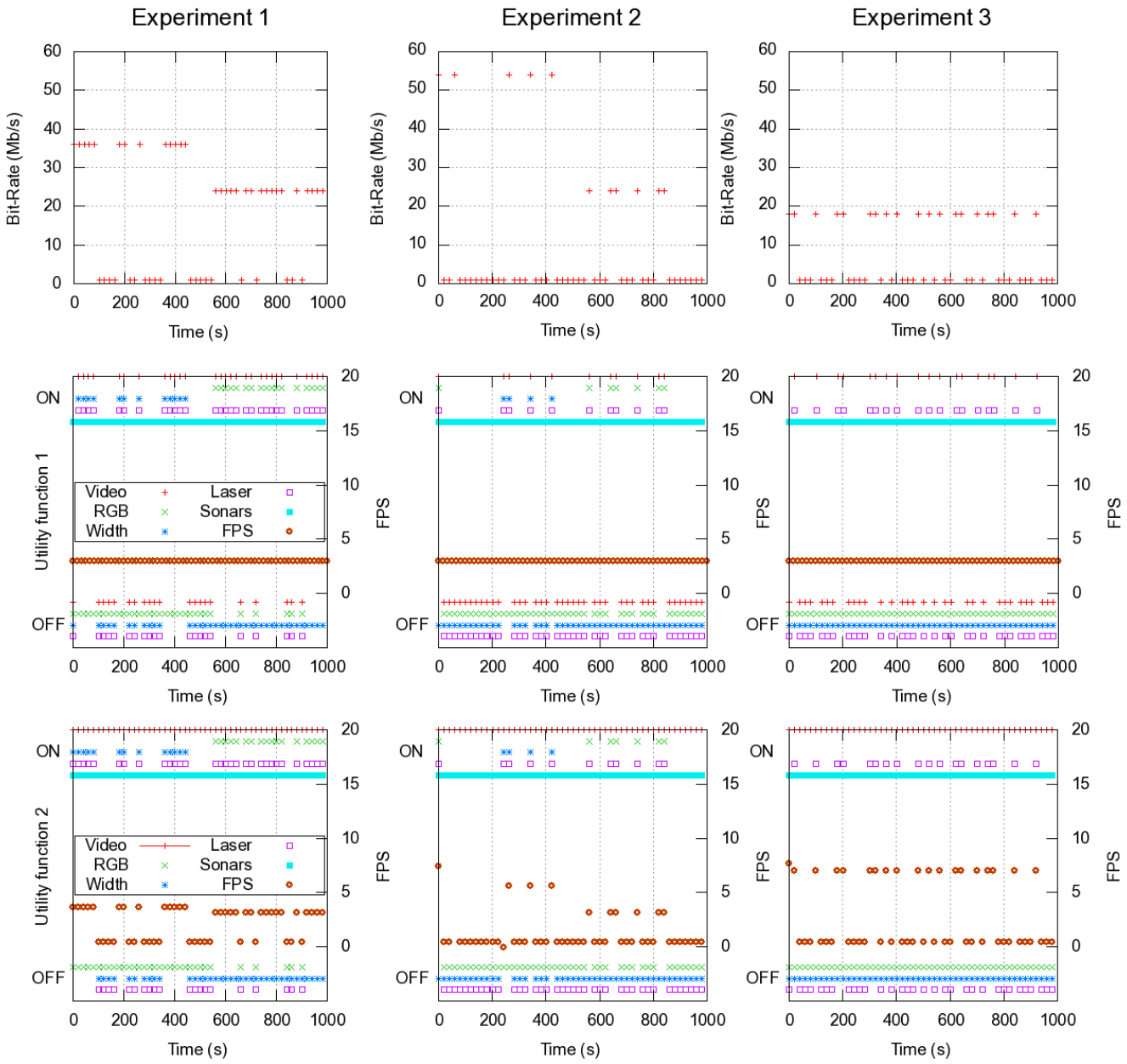

Figure 4. Results of three experiments with two different utility functions—$utility_1$ with $w_1 = 20/1280$, $w_2 = 10$, $w_3 = 20$, $w_4 = 10$; and $utility_2$ with $w_1 = 40$, $w_2 = 120$, $w_3 = 80$, $w_4 = 20$ and $w_5 = 20$. (These weights reflect the expected relevance of the various bandwidth consumers). Width off means $width = 640$, and width on means $width = 1280$.

B. Overhead

We measured the overhead of the autonomic bandwidth management solution by using the TCP/IP monitoring tool `tcpdump` [21] to keep track of the network traffic between a computer on-board the mobile robot and a remote workstation running the GPAC autonomic manager. Table III shows this overhead both as an absolute figure and as a percentage of the minimum bandwidth available between the mobile robot and the fixed control workstations (i.e., 1 Mbps), when the autonomic MAPE loop is executed once every second. The two rows in the table correspond to two configurations of the wireless connection between the on-board computer and the fixed workstation. The first row presents the overhead when the SOAP_IO_KEEPALIVE flag of the connection was not set. In this case, the underlying gSOAP framework [22] initiated a new connection for each web method call between the autonomic manager and the on-board computer, old connections being closed later.

Table I
RESULTS WITH UTILITY FUNCTION 1

Bandwidth (Mbps)	Sonars	Laser	Video	RGB	Width	FPS
1	on	off	off	–	–	–
2	on	on	off	–	–	–
6	on	on	off	–	–	–
9	on	on	on	off	640	3.00
12	on	on	on	off	640	3.00
18	on	on	on	off	640	3.00
24	on	on	on	on	640	3.00
36	on	on	on	off	1280	3.00
48	on	on	on	off	1280	3.00
54	on	on	on	off	1280	3.00

Table II
RESULTS WITH UTILITY FUNCTION 2

Bandwidth (Mbps)	Sonars	Laser	Video	RGB	Width	FPS
1	on	off	on	off	640	0.42
2	on	on	on	off	640	0.14
6	on	on	on	off	640	1.84
9	on	on	on	off	640	3.12
12	on	on	on	off	640	4.40
18	on	on	on	off	640	6.96
24	on	on	on	on	640	3.17
36	on	on	on	off	1280	3.66
48	on	on	on	off	1280	4.94
54	on	on	on	off	1280	5.58

Note that this behaviour was noticed thanks to the use of `tcpdump`, so in a second suite of experiments we set the SOAP_IO_KEEPALIVE flag to ensure that the connection between the autonomic manager was reused across multiple web method calls. While this improvement yielded an almost 20% reduction in the bandwidth overhead, the overhead when the default configuration was used (i.e., 3.9% overhead at the lowest possible bandwidth) was already acceptable. Note that for increased accuracy, this overhead can be taken into account in eq. (3).

Table III
AUTONOMIC MANAGER OVERHEAD

Case	Send bytes/s	Receive bytes/s	Total bytes/s	Overhead (at 1Mbps)
No Keepalive	2847	2267	5114	3.9%
Keepalive	2313	1809	4122	3.14%

VI. CONCLUSION AND FUTURE WORK

We integrated a mobile robot and a simulation of this robot with the GPAC autonomic computing framework. This enabled us to build a MAPE autonomic computing loop for the self-optimising management of the bandwidth between the on-board robot devices and the fixed workstations used to control this robot. Experiments were carried out to evaluate the effectiveness and the overhead of the solution.

There are several lines of work we would like to pursue in the future. We are planning to extend the use of the approach described in the paper to other types of robot resources (e.g., power usage). Additionally, we would like to improve the execution of the MAPE autonomic computing loop so that it is executed (and adds to the communication between the on-board robot laptops and the control workstations) only when a "significant" change occurs in the relevant robot parameters.

REFERENCES

[1] R. Sanz, A. Hernando, C. Martínez, and I. López, "Wrapping a Mobile robot with RT-CORBA," in *VIII IFAC Symposium on Robotic Control (SYROCO'06)*, September 2006.

[2] A. Hernando, "Versión RT-CORBA del servidor Pioneer 2AT-8," Master's thesis, ETSII, Universidad Politécnica de Madrid, October 2005.

[3] Y. Meng, "A Dynamic Self-Reconfigurable Mobile Robot Navigation System," in *Proc. of the 2005 IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics*, July 2005.

[4] B. MacDonald, B. Hsieh, and I. Warren, "Design for Dynamic Reconfiguration for Robot Software," in *2nd Intl. Conf. on Autonomous Robots and Agents*, Palmerston North, New Zealand, December 2004.

[5] "An architectural blueprint for autonomic computing," IBM, Tech. Rep., 2003.

[6] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing—degrees, models, and applications," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–28, 2008.

[7] R. Calinescu, "Model-driven autonomic architecture," in *ICAC '07: Proc. of the Fourth Intl. Conf. on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, p. 9.

[8] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das., "Utility Functions in Autonomic Systems," in *First Intl. Conf. on Autonomic Computing*. IEEE Computer Society, 2004, pp. 70–77.

[9] *Pioneer 3 & Pioneer 2 H8-Series Operations Manual, version 3*, ActivMedia Robotics, August 2003.

[10] *Common Object Request Broker Architecture (CORBA/IIOP)*, 3rd ed., Object Management Group (OMG), January 2008.

[11] K. Sevcik, "Interfacing with the Sick LMS-200," web page. [Online]. Available: http://www.pages.drexel.edu/~kws23/tutorials/sick/sick.html

[12] R. Calinescu, "General-purpose autonomic computing," in *Autonomic Computing and Networking*, M. K. Denko, L. T. Yang, and Y. Zhang, Eds. Springer, 2009, pp. 3–30.

[13] E. Werkman, B. van Schoonhoven, M. de Jonge, and E. Matthijssen, "Development of autonomic management solutions for the military application domain," in *5th IEEE Intl. Conf. on Engineering of Complex Computer Systems*. IEEE Computer Society, 2010, pp. 14–20.

 ISBN: 978-1-61208-109-0

[14] R. Calinescu and M. Kwiatkowska, "Using quantitative analysis to implement autonomic IT systems," in *Proc. of the 31st Intl. Conf. on Software Engineering (ICSE'09)*, 2009, pp. 100–110.

[15] ——, "CADS*: Computer-aided development of Self-* systems," in *Fundamental Approaches to Software Engineering (FASE 2009)*, ser. Lecture Notes in Computer Science, M. Chechik and M. Wirsing, Eds., vol. 5503. Springer, March 2009, pp. 421–424.

[16] R. Calinescu, "Reconfigurable service-oriented architecture for autonomic computing," *Intl. Journal On Advances in Intelligent Systems*, vol. 2, no. 1, pp. 38–57, June 2009.

[17] G. Tesauro *et al.*, "A multi-agent systems approach to autonomic computing," in *AAMAS '04: Proc. of the Third Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 464–471.

[18] J. O. Kephart and R. Das, "Achieving self-management via utility functions," *IEEE Internet Computing*, vol. 11, pp. 40–48, 2007.

[19] Y. Qiu and P. Marbach, "Bandwidth allocation in ad hoc networks: A price-based approach," in *IEEE INFOCOM*, 2003, pp. 797–807.

[20] I. N. Kassabalidis *et al.*, "Intelligent routing and bandwidth allocation in wireless networks," in *Proc. NASA Earth Science Technology Conf.*, College Park, MD, 2001.

[21] Lawrence Berkeley National Laboratory (LBNL), "tcpdump," web page. [Online]. Available: http://www.tcpdump.org/

[22] R. A. van Engelen, "gSOAP toolkit," web page. [Online]. Available: http://gsoap2.sourceforge.net/

 ISBN: 978-1-61208-109-0