The RAMON module: architecture framework and performance results

A. Roveri[1], C. F. Chiasserini[2], M. Femminella[3], T. Melodia[1], G. Morabito[4], M. Rossi[5], I. Tinnirello[6]

(1) University of Rome La Sapienza; (2) Polytechnic of Turin; (3) University of Perugia; (4) University of Catania; (5) University of Ferrara; (6) University of Palermo

Abstract. A design study of a Re-configurable Access Module for Mobile Computing Applications is described. After a presentation of its cross-layered architecture, Control Parameters (CPs) of the module are introduced. The set of CPs both describes the functional state of the communication process in relation to the time-varying transport facilities and provides, as input of suitable Algorithms, the control information to re-configure the whole protocol stack for facing modified working conditions. The paper also presents the structure of the simulator realized to demonstrate the feasibility of the design guidelines and to evaluate reconfigurability performances.

1 Introduction

This paper presents a first insight in the design of a Re-configurable Access module for MObile computiNg applications (called RAMON in the sequel) and discusses some preliminary results of a feasibility study of this module which is being carried out within an Italian research program[1], with the participation of six academic research groups[2].

The framework of RAMON is given by mobile users in a TCP/IP environment (*mobile computing*). The mobility is allowed in different communication environments (*Reference Environment, RE*). These REs are identified in a WAN cellular system (e.g., UMTS or GPRS) and in a local area access system (e.g., IEEE 802.11 or Bluetooth).

In wireless communications, much effort will have to be put in the ability to re-configure all layers of the communication process to the time-varying transport facilities, in order to assure the respect of QoS *(Quality of Service)* requirements. The adaptability to different communication environments led us to overtake the OSI layered approach and to adopt the idea of *cross-layering design*, which appears to be particularly appealing to enable dynamic optimization among all layers of the communication process. The idea has already been

[1] The RAMON project was partially funded by MIUR (*Italian Ministry of Research*).

[2] The address of the reference Author is: *roveri@infocom.uniroma1.it*

presented in [1], [2] as a design methodology for adaptive and multimedia networks. RAMON provides a first example of use of this methodology.

The remaining of the paper is organized as follows. In Section 2 we describe the RAMON Functional Architecture. Section 3 defines the relevant *Control Parameters* that constitute the basis of the re-configuration action and the main algorithms which perform it. Section 4 presents the overall RAMON simulator developed in the `ns`-2 framework [3] to demonstrate the feasibility of the design guidelines and to evaluate the reconfigurability performances, while Section 5 contains some samples of performance results.

2 The Architecture

RAMON allows the deployment of control algorithms which are independent of the specific RE considered. A *mobile station* (MS), equipped with RAMON, becomes a *virtual MS*, i.e. an MS which supports *abstract functionalities* common to all the REs. Such functionalities are obtained from the *specific functionalities* of each RE, by means of *translation entities*, as described below.

Figure 1 shows the RAMON functional architecture. It includes a *Common Control* plane (*CC-plane*) and an *Adaptation* plane (*A-plane*), which is divided in as many parts (*A/RE-i*) as the number of different REs involved. In the case of Figure 1 the number of different REs is let equal to two for the sake of simplicity. Below the A-plane, *Native Control* functionalities (*NC-plane*) for each RE (denoted as *NC/RE-i* for the *i*-th RE) are located. An RE-independent service development *API* (*Application Programming Interface*) is offered to the overlying Application layer.

CC-plane functions, which are also RE-independent, are grouped in three *functional sets*, according to the classical model adopted for wireless communications: i) *Radio Resource Control* (*RRC-c*); ii) *Session Control* (*SC-c*); iii) *Mobility Management* (*MM-c*).

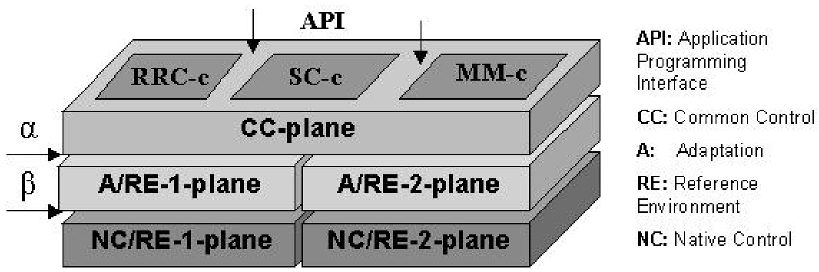

Fig. 1. Overall system architecture.

The A-plane translates: 1) *Primitives* exchanged between the CC-plane and the NC-plane for each RE; 2) *Control Parameters* (CP) data passed from the *Native User* plane (*NU-plane*) to the CC-plane.

Two different types of interfaces can be identified: i) the α *interface*, between the CC-plane and the A-plane; ii) the β *interface*, between the A-plane and the NC-plane of each RE. Figure 2 completes Figure 1. In its right part the CC-plane and the A-plane are depicted, with the relevant α and β interfaces. In its left part, which is concerned with two generic REs, the relevant NC-planes and NU-planes are shown.

Figure 2 helps us showing the way the SC-c, MM-c and RRC-c functionalities interact with the corresponding *SC-i*, *MM-i* and *RRC-i* (i=1,2) of the two REs through the translation performed by the A planes (A/RE-1, A/RE-2). The relations between the NC-planes and NU-planes are highlighted and particularly how the SC-i, MM-i and RRC-i functionalities are related with *Physical* (PHY-i), *Medium Access Control* (MAC-i) and *Radio Link Control* (RLC-i) functions is shown. Finally, *Applications*, *TCP/UDP* and *IP* layers, which are part of the NU-plane, directly interact with the CC-plane.

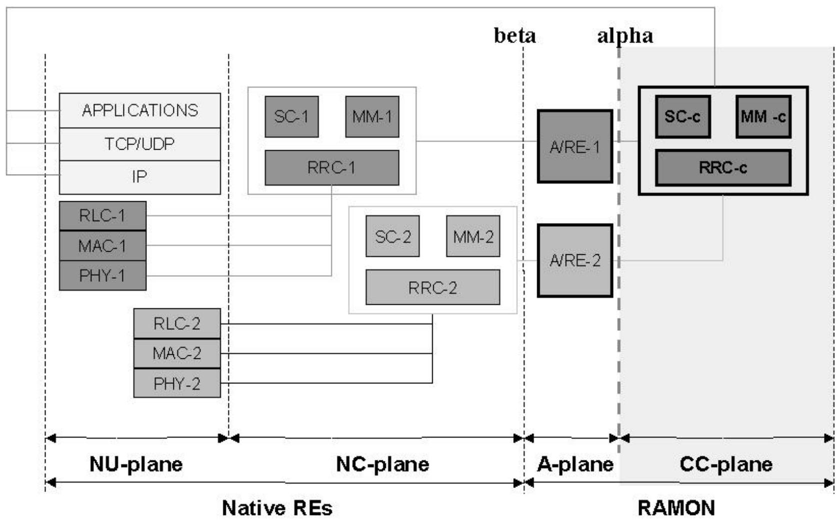

Fig. 2. Basic interactions among Control and User planes.

3 Reconfigurability Parameters

The main idea is to dynamically adapt the whole stack to the perceived QoS by means of adequate control actions. QoS measurements are passed to the RAMON entity by means of CP data.

Cross-layer interaction requires control information sharing among all layers of the protocol stack in order to achieve dynamic resource management. It may be performed in two directions: i) *upward information sharing*: upper layers parameters may be configured to adapt to the variable RE characteristics; ii) *downward information sharing*: MAC, RLC and PHY layer parameters can adapt to the state of Transport and Application layer parameters. Thus, e.g., the behavior of the TCP congestion window may be forced to adapt to the time-varying physical channel characteristics (upward), while lower layer parameters may be re-configured, with the aim of serving distinct applications with different QoS requirements (downward).

As shown in Figure 3, CP data flow from all layers of the NU-plane of each RE and are processed by *Algorithms* running in the CC-plane. The output of the Algorithms consists of *Command Primitives*, which are passed to all layers of the protocol stack. For lower layers, which are RE-dependent, these primitives have to be interpreted by the pertinent A-plane; for higher layers, which are typical of the Internet stack, the primitives can act without any mediation.

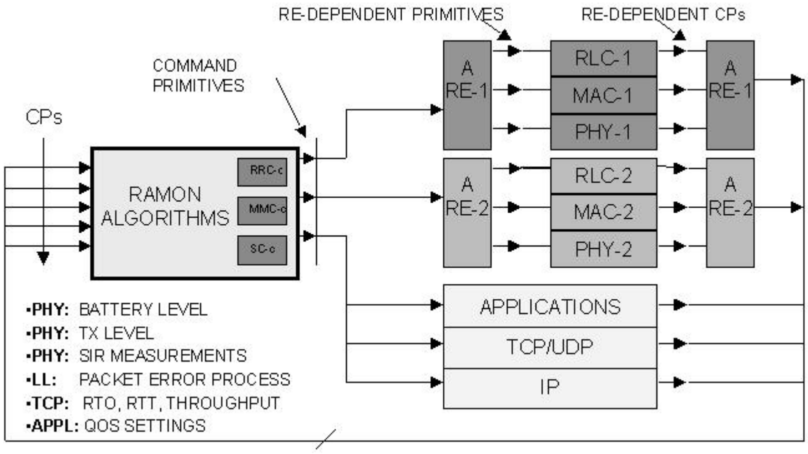

Fig. 3. Information flows in RAMON.

The effect of the Command Primitives, in concurrence with the variations of the RE transport characteristics, gives rise to modifications in the perceived QoS. These modifications cause corresponding change of the CP data. The CC-plane performs re-configuration actions only if they entail appreciable improvements in the QoS perceived.

The most important Algorithms adopted in the RAMON design are:

– *The Handover* Algorithm;
– *Session Control* Algorithm;
– *QoS & MM* Algorithm;

– RRC Algorithm for *Error Control;*
– RRC Algorithm for *Resource Sharing Control.*

The first two algorithms will be analyzed respectively in Sections 3.1 and 3.2; the other three will be briefly described in the following.

QoS and MM capabilities are carried out by means of the Mobile IP protocol, improved with an *Admission Control* function. This last is called *GRIP* (*Gauge and Gate Realistic Protocol*), is described in [7] and operates in a *Differentiated Services* framework. In addition, we assumed that a *micro-mobility strategy* [8] is adopted within the wireless domain: the scope is to hide local handoffs from Home Agent/Correspondent Nodes, thus reducing handoff latency.

To preserve information integrity of packet transmission over the radio channel while meeting the desired QoS and energy constraints, an algorithm for Error Control has been developed within the RRC functional set, which is suitable for a UMTS environment: it provides optimal operational conditions and parameter setting at the RLC layer. The performance results of the application of the algorithm to the UMTS RE are being published [10] and a module which implements it in the RAMON overall simulator has been developed.

An open, re-configurable scheduling algorithm called CHAOS (CHannel Adaptive Open Scheduling) has been defined, which is part of the RRC-c functional set. The Algorithm defines a scheduling strategy for efficient resource sharing control. It is adaptive to *traffic conditions* and *physical channel conditions*. Different CPs may be chosen to represent both variables. Traffic conditions may be represented by transmission buffer occupancy or queue age associated to packets in the transmission buffers; physical channel conditions may be expressed by averaged *Signal to Interference Ratio* (SIR) and/or *Packet/Frame Error Ratio*. The basic CHAOS principles can be applied in a RAMON module residing in a VBS managing radio resources of different REs. The specific adaptations studied for UTRA-TDD and Bluetooth (whose description can be found in [11]) correspond to A-plane entities of the RAMON architecture.

3.1 Handover Algorithm

RAMON includes an abstract handover algorithm, running at the Virtual MS (VMS), based on the virtualization of the functions necessary for Mobility Management. This operation allows making handover services programmable and independent of the underlying technologies.

In a re-configurable wireless scenario, the selection of the "best" access point at any moment is not simply related to the physical channel quality, but implies also a technology choice as a trade-off between performance, cost, power consumptions, etc. Moreover, the roaming across different systems requires solving addressing and authentication problems, maintaining a low latency to prevent the adverse effects of TCP congestion control.

In order to develop a platform-independent handover algorithm, similarly to [4], [5] we have decoupled these problems. By hiding the implementation details of the MM algorithms from handover control systems, the handover decision can be managed separately from handover execution, so that the same detection mechanisms can interface with many different access networks.

The algorithm is based on a generic *Mobile Controlled Handover* (MCHO) style: handover decisions and operations are all done at the VMS. RAMON keeps track of a list of *Virtual Base Stations* (VBS) in its coverage area. The rationale is that some REs do not provide MCHO capabilities. Therefore, the entire RE is considered as a unique VBS. Conversely, for MCHO systems, a VBS coincides with a BS.

Periodically, for each VBS, the VMS collects CPs as QoS measures (possibly querying the VBS) and estimates cell load condition. According to the definition criterion of the best VBS, if the best serving VBS is not the one in use for a given period of time, the VMS hand-offs to it.

We can identify three main logical functional blocks: i) the *user profile specification*, which re-configures the decision metrics, according to the user requirements; ii) the *measurement system*, based on system-dependent available functions and signaling; iii) the *detection algorithm*, which compares abstract performance metrics computed on the basis of the measurements and of the user profile.

The cost of using a BS_n at a given time is function of the available *bandwidth* B_n, the related *power consumption* P_n, the *cost* C_n of the network the BS_n belongs to. While the last two parameters are easy to compute (mobile host battery life and service type cost), the bandwidth parameter is more complex. In order to account for the real available bandwidth, channel quality perceived by the MS and cell occupancy status have to be considered.

The cost function of the BS_n is then evaluated as:

$$f_n = w_b \cdot N(1/B_n) + w_p \cdot N(P_n) + w_c \cdot N(C_n) \tag{1}$$

where w_b, w_p and w_c are the weights of each parameter, which sum to 1, and $N(x)$ is the normalized version of the parameter x. The reciprocal of the bandwidth is considered in order to minimize the cost function of the best access BS.

Weights can be modified by the user at run-time, and can be set to zero for those parameters that are not of concern to the user: e.g., if high performance has to be pursued, we can assign $w_b = 1$, $w_p = 0$ and $w_c = 0$. Thus, by minimizing the cost function, we achieve load balancing across different REs.

3.2 Session Control Algorithm

Due to the low Mobile IP performance, the MS migrating from a RE to another may be unreacheable for time periods of the order of seconds, which considerably impacts the TCP operation. In particular, several consecutive RTOs (Retransmission Time Out) occur. This results in a very low value of the *Slow Start Threshold (ssthresh)* which, upon each RTO expiration, is updated as follows:

$$ssthresh = \max(cwnd/2,\ 2 \cdot MSS) \tag{2}$$

where *cwnd* denotes the current *Congestion Window*. Therefore, as the inter-RE handover is successfully completed, the sender immediately enters the *Congestion Avoidance* phase and *cwnd* increases slowly. This causes low throughput performance.

In order to solve this problem, the TCP sender implementation has been modified, still maintaining compatibility with standard TCP implementations. We observe that, after an inter-RE handover, the connection path may change dramatically. Consequently, what the TCP sender learned in terms of available bandwidth and estimations of the RTT (Round Trip Time) and RTO is not valid anymore. Based on the above observation, after the inter-RE handover is completed, the TCP sender resets its internal parameters (i.e., *ssthresh*, *Smoothed RTT*, and *RTO*), and enters into the so-called *Fast Learning* phase during which it rapidly estimates the available bandwidth and the RTT characterizing the connection in the new RE. Moreover, in order to avoid useless segment transmissions which would only result in power consumption, the TCP sender *stops* transmitting any data as the handover begins and *resumes* as the handover is successfully completed. The information about the beginning and the end of handovers is provided by command primitives.

The above Algorithm [6], which will be referred to as TCP-RAMON in the sequel, has been integrated in the overall RAMON simulator, and is effective when the MS, acting as a mobile host, is the TCP sender. Otherwise, a similar behavior can be obtained as follows. Before initiating the handover, the MS generates an ACK which informs the TCP sender that the *Receiver Window* (*rwnd*) is equal to 0. This results in a *freezing* of the TCP sender and avoids consecutive RTO expirations. When the handover is completed, the TCP receiver generates a new ACK with the original value of *rwnd*: this occurrence resumes the communication.

4 The Simulative Approach

In this section the RAMON simulator (Figure 4) is described with reference to UMTS-TDD and Bluetooth REs. An 802.11 implementation has also been developed. The simulator is based on the `ns` package [3] (ver. 2.1b7a). Specifically, modifications have been carried out to the Wireless Node object (*MobileNode* class). One protocol stack for every simulated RE is created in the same MobileNode object.

The RAMON module and the A-plane modules are interposed between Agents (RTAgent, MIP Agent, etc.) and radio access layers (LL, MAC, etc.). On the left part of Figure 4 the UMTS-TDD protocol stack is shown with its specific Adaptation plane (*A/RE-UMTS* in figure), where as the Bluetooth stack is on the right side with its A-plane (*A/RE-BT*). On top of the stacks the RAMON module is directly linked to the TCP (or UDP) layer, to the MIP layer (*Mobile IP*) and to the *NOAH* routing Agent (NOn AdHoc routing agent [13]). The UMTS-TDD protocol stack has been developed for this project. The Bluetooth protocol stack is derived from the `ns` *Bluehoc* simulator [12] and modified to

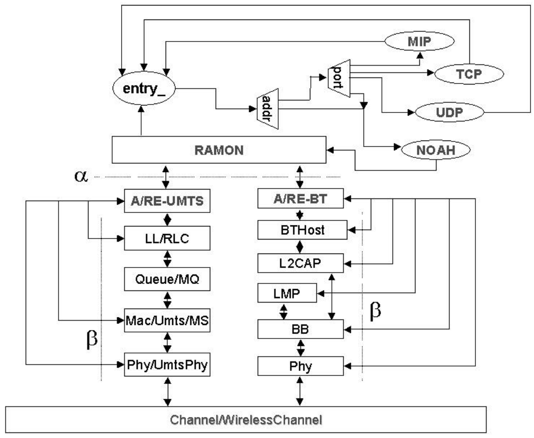

Fig. 4. Overall simulator architecture.

adapt the Bluetooth node object *BTnode* to the `ns` MobileNode object. The α interface in Figure 4 defines the messages exchanged between CC-plane and A-planes. The functions of the α interface can be related to the Command Primitives and the parameters they exploit to the CPs. A list of the most important of these functions follows:

- `get(parameters_name)`: gets the `parameters_name` parameter value.
- `attach()`: is used to create an *IP context* and to register the node to the *Foreign Agent* (*FA*).
- `detach()` deletes registration from the FA.
- `monitor(RE)`: returns a quality metric for the requested RE.
- `set(parameters_name,value)`: sets the `value` value to `parameters_name` parameter.
- `send(packet,options)`
- `receive(options)`

In particular, the "`get`" function is used to read the CPs from the A-planes for optimization purposes. CPs originated at different layers can be associated to different optimization tasks: i) physical layer CPs are related to power control and battery life saving; ii) link layer CPs are directly connected to link reliability and packet delay; iii) transport layer CPs are directly related to the QoS perceived from the user application in terms of packet delay and throughput. The "`set`" function is needed to pass some values to the A/RE-i modules: in fact, with this primitive, some CPs can be modified by the CC-plane. Both the "`set`" and the "`get`" functions are also defined for the interactions between the CC-plane and RE-independent upper layers protocols. The other functions defined above are Command Primitives (for the α interface).

Two more primitives (named respectively `start_handover_notification()`

and `end_handover_notification()`) are defined for interaction between the CC-plane and the transport layer. The first one notifies the TCP about the beginning of the handover while the second one notifies the end of it.

The most important CPs passing through the α interface are briefly discussed in the following for the main layers interacting with RAMON.

Physical layer. CPs detectable at such layer are strongly dependent on the considered RE. A very useful CP is the *battery level* (`BAT_LEVEL`), used to maximize the MS life, e.g., by attaching the MS to the less power demanding RE. Moreover, *Transmitted Power* (`TX_POWER`) and *SIR (Signal to Interference Ratio)* (`SIR`) measurements, if available, are used to detect temporary link failures or high interference conditions: e.g., if the physical channel in a given instant is detected to be in an interference prone situation, the Link layer can be set in a "wait" status to avoid energy wastage.

Link layer. A description of the *error process* at such layer is significant in order to characterize the packet error process affecting the Application layer. Such a description can be achieved by translating the Link layer packet success/failure transfer process into a simple *Markov Model* [14], which is analytically tractable and RE independent; hence RE independent algorithms can easily be deployed which exploit such model.

The CPs for the models are: *loss probability* at the link layer (`P_LOSS_LINK`), *link layer packet size* (`LINK_PACKET_SIZE`), *average burst error length* at link layer (`BURST_ERR`) and *average link packet error probability* (`ERR_PR`). Other useful CPs are the *maximum number of retransmissions per packet* allowed in the link layer (`MAX_LINK_RETR`) and the *average number of retransmissions per correctly transmitted packet* (`AVG_LINK_RETR`); the last one is related to the energy spent per useful information bit, i.e., the amount of energy wasted due to Link layer retransmissions.

Transport layer. At this layer, the MS should be able to collect information regarding the packet *delay/error* process. When the TCP protocol is considered, in addition to the packet error rate, it communicates to RAMON the average and instantaneous *Throughput* value (`AVG_TH` and `ACT_TH`), the estimated *Round Trip Time* (`RTT`) and the actual `RTO`. These CPs are used to control the behavior of TCP state variables and to avoid their incorrect setting as packet errors occur. In particular, it is well known that TCP reacts to packet losses as if a congestion had occurred, i.e., by reducing the maximum flow that the sender is allowed to put into the network at a time. However, in a wireless environment this is not the correct way to proceed because errors affecting a wireless link are very often transient and do not represent at all a measure of network congestion. On the contrary they are a measure of the temporary link QoS. Moreover, performance measured at this level is directly related with the QoS perceived by the user, so it can be weighted in a cost function in order to derive an estimate of the actual QoS. Such an estimate is then compared against the required QoS and used by RAMON algorithms in their decisional tasks.

5 Performance issues

The most meaningful simulated situations are:

1) Forced inter-RE Handover.
2) User-driven inter-RE Handover;

These scenarios are relevant in the RAMON context because they easily show how a Common Control Plane (RE-independent) can react to variations in the REs it handles, perceived by means of Control Parameters, for optimization purposes. In the first scenario the VMS loses connectivity and is forced to attach to another RE to maintain session continuity. In the second one the VMS is in the coverage area of two different REs (UMTS and 802.11) at the same time and chooses the best RE evaluating a cost function as described in Section 3.1.

5.1 Forced Handover

In this Section some simulation results relevant to an inter-RE forced handover are presented.

The scenario involves a UMTS BS, which is the MIP *Home Agent* (HA), and a Bluetooth BS, which has the role of *Foreign Agent* (FA). A Forced Handover between the former and the latter is simulated.

Figure 5 shows the temporal behaviour of the TCP sender *cwnd* when TCP NewReno is used and when the modified TCP-RAMON is used. When TCP NewReno is used, multiple RTOs expire during the handover and lead to the ssthresh to its minimum value. Thus, when the handover ends, the TCP enters in the *congestion avoidance* phase and thus its cwnd increases slowly.

When TCP-RAMON is used, as soon as the CC-plane detects a loss of connectivity, a `start_handover_notification()` primitive is issued to the TCP layer. This freezes the value of ssthresh to the one determined by the last congestion event occured. Accordingly, timeouts expiring during the handover do not impact the value of ssthresh. Once the `end_handover_notification()` is received the value of sstresh is set equal to the one determined by the last congestion event. Thus, TCP enters the slow start phase (it does not enter the congestion avoidance phase until cwnd reaches the ssthresh value). In these simulations, buffers are considered to be infinite and the initial ssthresh value has been set to 20 which is a typical value in several TCP implementations.

5.2 Handover Customization

In this Section we present some simulation results relevant to user-driven handovers. Our purpose is to demonstrate how user profile specifications affect handover trigger and detection phases. In fact, the optimization of different performance figures requires different choices in terms of RE attachment/detachment policy. For example, if user main goal is cost saving, connections to low-price access points have to be maintained as long as possible, even if better transmission

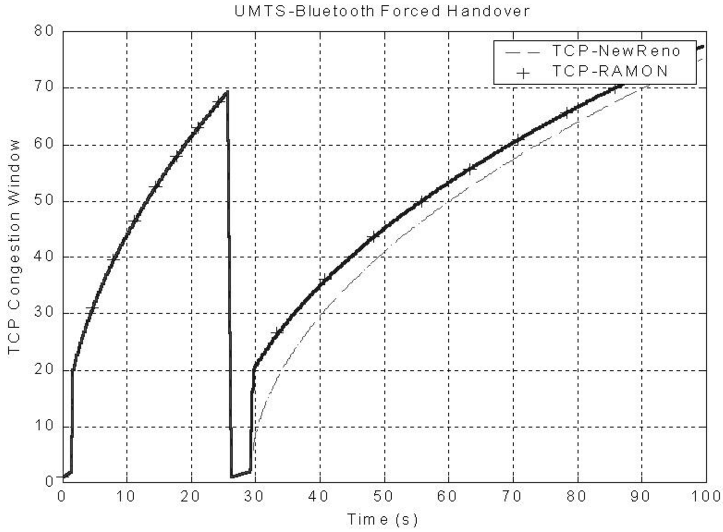

Fig. 5. UMTS-Bluetooth Forced Handover.

conditions (in terms of bandwidth or channel quality) towards other stations are available.

In our simulations we consider an area in which heterogeneous radio access technologies (namely UMTS and 802.11), experiencing different traffic conditions, overlap. In particular, we have considered a simple network topology: four 802.11 BSs (referred to as BS_2, BS_3, BS_4, BS_5) are placed at the vertices of a square and a UMTS BS (BS_1) is placed in the centre. The side of the square is set to 600m, while the coverage areas of the 802.11 and UMTS BSs are set respectively to 200m and 1000m. Channel rate in 802.11 cells is set to 2 Mb/s. 802.11 BSs have a different number of attached users: four mobile nodes are connected to BS_2 and BS_4, while nine mobile users are connected to BS_3 and BS_5.

In the simulation run, a RAMON mobile node, involved in a 6 Mbytes file transfer from the fixed network, moves clock-wise at 6m/s along the square starting from a vertex, for example from BS_2. Although during the simulation time RAMON node is covered by BS_1, handover to 802.11 BSs can be performed in order to save power or reduce cost. Nevertheless, an handover towards 802.11 BSs implies a lower amount of available bandwidth because of their load conditions. The required user trade-off is expressed by the decision metric (Section 3.1) settings. We compute such a metric considering distance, price and bandwidth offered by each BS. In order to make different system parameteres comparable, we normalize each parameter as follows: distance is expressed as the ratio between actual distance and maximum coverage radius, price and bandwidth are normalized with respect to the maximum inter-RE values. In particular, we assume that the cost to transfer 1 kbyte of data is 1 for UMTS, 0.5 for 802.11. In our simulations, we set the metric distance weight to 0.5 and we observe

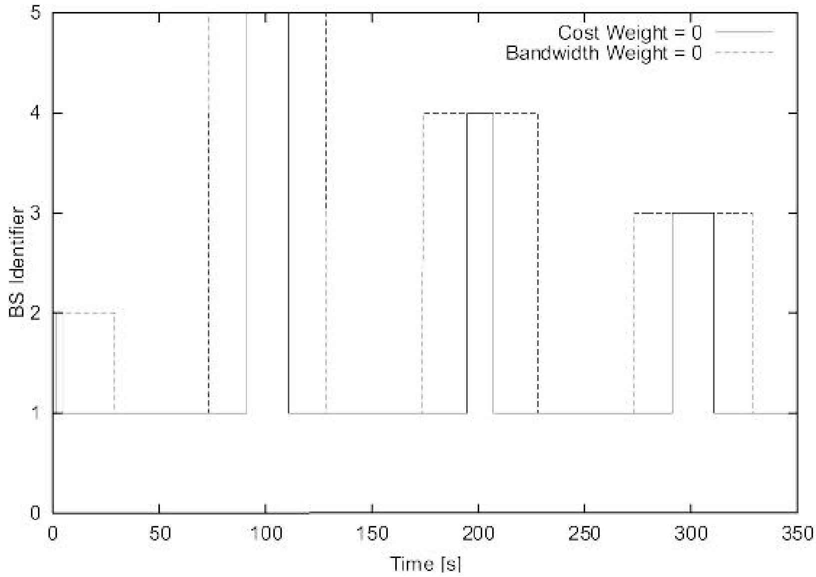

Fig. 6. Selected BS vs simulation time.

performance results varying the cost and bandwidth weights according to the relation: $w_b = 0.5 - w_c$. Consider preliminarly the extreme situations in which we want to minimize the file transfer-cost regardless of the transfer-time ($w_b = 0$) or, conversely, we want to minimize the transfer-time regardless of the cost ($w_c = 0$). Figure 6 shows the handover trigger and decision policies adopted in the cosidered extreme cases. We report the identifier of the selected BS versus the simulation time. In the w_b situation, connection to WLAN is kept as long as possible, since this access technology is the cheapest (handovers are triggered only when the RAMON node moves outside the coverage area of an 802.11 BS). In the w_c case, the RAMON node remains connected to UMTS for a greater time (switch to WLAN occurs only when the distance from the relevant BS is very small). Note that, since our metric accounts for both distance and load, the permanence time in 802.11 BSs is not constant and depends on the BS considered.

The time and the cost resulting for the considered 6 Mbytes file transfer is reported in Figure 7 versus the w_c setting. This figure shows that, by changing the weights of the cost function, the RAMON user can effectively configure its optimal cost/performance trade-off.

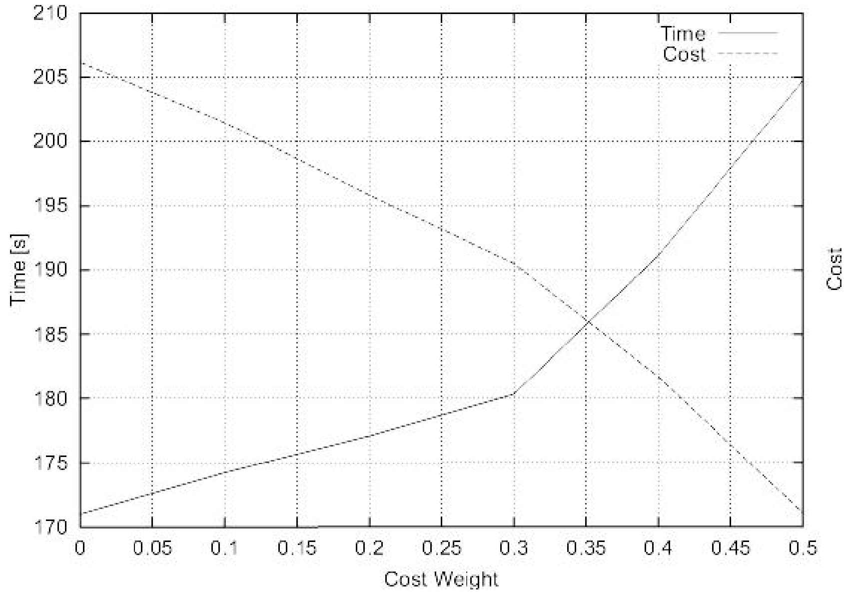

Fig. 7. Time and Cost Trade Off.

6 Conclusions

In this paper the guidelines adopted for the design of a module able to reconfigure different mobile communication environments for computing applications are introduced. The paper mostly concentrates on architectural issues and on control information flows. In particular, the Algorithms that constitute the processing core of the module are briefly described. The integrated RAMON simulator is described and some performance results are finally shown.

7 Acknowledgments

The authors would like to thank all the researchers of the Polytechnic of Turin and of the Universities of Catania, Ferrara, Palermo, Perugia and Rome "La Sapienza" who participated to the RAMON project and, notwithstanding their important contribution, do not appear as authors of this work.

References

1. Z.J. Haas, "Design Methodologies for Adaptive and Multimedia Networks," IEEE Communications Magazine., Vol. 39, no. 11, November 2001, pp 106-07.
2. T.S. Rappaport, A. Annamalai, R.M. Buehrer, W.H. Tranter, "Wireless Communications: Past Events and a future Perspective," IEEE Commun. Mag., Vol 39, no. 11, May 2002, pp 106-07.

3. Network Simulator *http://www.isi.edu/nsnam/ns/*.
4. M. Stemm, R. H. Katz, “Vertical Handoffs in Wireless Overlay Networks,” ACM Mobile Networking (MONET) Vol 3, n 4, 1998, pp 335-350.
5. M.E. Kounavis, A. T. Campbell, G. Ito, G. Bianchi, “Design, Implementation and Evaluation of Programmable Handoff in Mobile networks,” ACM Mobile Networking (MONET), Vol. 6, Sept. 2001, pp. 443-461.
6. G. Morabito, S. Palazzo, M. Rossi, M. Zorzi, “Improving End-To-End Performance in Reconfigurable Networks through Dynamic Setting of TCP Parameters,” published in these Proceedings.
7. G. Bianchi, N. Blefari-Melazzi, M. Femminella, F. Pugini, “Joint support of QoS and mobility in a stateless IP environment,” IEEE GLOBECOM 2001, San Antonio, USA, November 2001.
8. A.T. Campbell et al., “Comparison of IP micromobility protocols,” IEEE Wireless Communications, February 2002.
9. M. Femminella, L. Piacentini, “Mobility Management in a Reconfigurable Environment: the RAMON Approach,” published in these Proceedings.
10. C. F. Chiasserini and M. Meo, “A Re-configurable Protocol Setting to Improve TCP over Wireless,” IEEE Transactions on Vehicular Technology (to appear).
11. A. Baiocchi, F. Cuomo, T. Melodia, A. Todini, F. Vacirca, “Reconfigurable packet scheduling for radio access jointly adaptive to traffic and channel,” published in these Proceedings.
12. Bluehoc Simulator *http://www-124.ibm.com/developerworks/opensource/bluehoc/*.
13. J Widmer, “Network Simulations for a Mobile Network Architecture for Vehicles,” *http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/*.
14. M. Zorzi, R.R. Rao, “Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks,” IEEE Personal Communications, vol. 6, Oct. 1999.