# A Cross-Layering and Autonomic Approach to Optimized Seamless Handover

G. A. Di Caro*, S. Giordano, *Member IEEE*†, M. Kulig†, D. Lenzarini *Member IEEE*§, A. Puiatti† and F. Schwitter†
*Istituto "Dalle Molle" di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, CH-6928 Manno, Switzerland
Email: gianni@idsia.ch
†University of Applied Sciences (SUPSI), Galleria 2, CH-6928 Manno, Switzerland
Email: {silvia.giordano,kulig,alessandro.puiatti,schwitter}@supsi.ch
§Forward Information Technologies SA, Galleria 2, CH-6928 Manno, Switzerland
Email: davide.lenzarini@forit.ch

***Abstract*— Performing global node mobility requires to support seamless vertical and horizontal handovers between different network providers and technologies. The optimization of the handover processes requires not only continuous network services, but also continuous service performance. In this paper we present a novel design approach based on autonomic components and cross-layer monitoring and control to optimize the performance of the WiOptiMo system, which provides seamless internetwork roaming by handling mobility at the application layer. We also distinguish from past work, in that we pragmatically followed the approach to rely only on existing technologies, deployed protocols and lightweight calculations, such that our system can be straightforwardly implemented as it is on most of the currently available mobile network devices. We report results from some simple real-world experiments showing the benefits of using a cross-layering approach. We also describe a first version of the new WiOptiMo based on the innovative design. Results from preliminary tests in real-world scenarios indicate the effectivness of our pragmatic approach. The system is still under development and testing, and we plan to integrate in it several autonomic components, which are presented and discussed in the paper.**[1]

## I. Introduction

In forthcoming scenarios for heterogenous mobile networks users can benefit of ubiquitous coverage by roaming between different access networks (*internetwork roaming*). The challenge consists in providing seamless and continuous connectivity with the QoS required by the user applications given the different characteristics (in terms of coverage, bandwidth, cost, etc.) of the different available networks. The optimization of the decisions and procedures of internetwork roaming is a key issue for the successful deployment of pervasive and ubiquitous network services. While several solutions have been proposed in the past for the internetworking roaming problem, usually they are either too complex to be efficiently implemented in real-world applications, or they assume the access to parameters and information which are hardly available given current standard protocols and devices.

The solutions we present in this paper, integrated in the *WiOptiMo* [1], [2] system, have been designed to overcome the drawbacks of previous work. WiOptiMo is a middleware system based on a pair of Client-Server modules at the application layer. These modules interface the communication between the actual Client and Server applications hiding the mobility to them, by letting them "believe" that they run on the same machine.

While these core characteristics of the WiOptiMo approach have been left unchanged, in this paper we present several improvements of the system in terms of self-tuning and adaptation of parameters and optimization strategies, collected information, and optimization of the adopted decision procedures on the basis on this same information. The proposed modifications are the result of a novel design approach based on the use of *autonomic* [3], [4] components and *cross-layer* [5], [6] monitoring and control. We claim that this is the way the go to deal effectively with the challenges posed by the dynamic, probabilistic, and technological aspects of the multiobjective optimization problem at hand. More specifically, the proposed novel approach, which is based on the collaboration among *physical*, *network* and *application* layers to act efficiently at the network layer, is expected to boost-up system performance, as well as to show a good level of *adaptivity*, which is a key component to properly act and react in the highly dynamic scenarios of interest. Furthermore, the architectural innovation is supported by autonomic components: the system is able to *self-configure* and to *self-optimize* its internal parameters when introduced in a novel hardware/network context, and in the future it is expected to use past experience to show anticipatory behaviors and/or learn about user preferences. As opposite to what hase been proposed in the past for the internetworking roaming problem, in the *WiOptiMo* system we pragmatically followed the approach to rely only on *existing technologies, deployed protocols and lightweight calculations*, such that our system can be straightforwardly implemented as it is on most of the currently available mobile network devices. Clearly, our approach has the "drawback" that we have to face challenging limitations in terms of information available to take statistically sound and optimized decisions, and in terms of complexity of the implemented algorithms.

This paper has to be seen as a bridge between the original WiOptiMo system, based on traditional design, and the new

[1]This work was partially funded by the CTI Swiss programme under the KTI-7640.1 ESPP-ES Optimised always-on solution project.

version of it, based on cross-layering and autonomic components. In fact, the new system is still under development and testing. Such that in this paper, we introduce the general motivations and characteristics of the novel approach, discuss the envisaged solutions, and report some experimental results supporting our point of view. We also describe in some details the current status of the new implementation.

The main contributions of the paper are: (1) the introduction of a novel cross-layering and autonomic approach for seamless and efficient internetwork roaming, (2) a set of experimental results from real-world scenarios providing a first validation of the soundness of the approach, and (3) the description of a beta implementation of the new WiOptiMo, based on a minimalist cross-layering and autonomic design to allow the effective and portable implementation on commercial platforms.

The rest of the paper is organized as follows. Next section reports a short summary of the general characteristics of the internetwork roaming problem and of the proposed solution approaches. Section III and its subsections describe the general architecture and the specific components of the original WiOptiMo system. Section IV provides the generalities about the cross-layering and autonomic design approach. Subsection IV-A discusses the proposed solutions for cross-layer monitoring and control at the physical and application layers. Subsection IV-B describes the autonomic components of WiOptiMo, while Subsection IV-C discusses the components of the system dealing with user interaction. Section V reports results of real-world experiments aimed at showing the effectiveness of application layer active measures to infer traffic load information. Section VI describes the new implementation of the Check Activity, one of the main components of WiOptiMo, redesigned according to the cross-layering and autonomic point of view. Finally, Section VII draws some conclusions and discusses future work.

## II. Definitions and General Characteristics of the Internetwork Roaming Problem

The problem of internetwork roaming is also referred to as the problem of network *handover* (or *handoff*). The switching between two different types of networks is called *vertical handover*, while *horizontal handover* refers to the case of networks of the same type. The handover can be either *soft* (or *alternative*) when it is executed for the sole purpose of *optimization* of the connection cost or QoS, or *hard* (also termed *imperative*), when it is executed due to imminent or present loss of connectivity.

The handover of the mobile terminal (MT) is *network executed* if it is done by the network connection point (e.g., as it is the case between UMTS/GSM/GPRS cells). *Mobile executed handover* is the case in which the handover decision is autonomously taken by the MT. This is the modality prescribed by the currently deployed 802.11 standard for WLANs and this is the main focus of our work. For mobile executed handovers the strategies for horizontal handovers between WLANs, and vertical handovers from WLAN to WWAN and vice versa, have slightly different characteristics due to the existing differences in the information available to the MT at decision time and in terms of coverage, bandwidth, cost, and signal strength between the two different types of networks.

Building-up on previous work for handover in GSM networks, the majority of the approaches for horizontal handover in WLANs reason on the "quality" of a connection making use of simple *threshold-based schemes* usually including *hysteresis margins* [7] and *dwell timers* to enhance overall robustness (e.g., see [8]). The quality of a connection can be expressed by means of any desired combination of metrics related to PHY measures of the quality of the received signal and/or to MAC/TCP/APP measures of the available bandwidth. It is common practice to restrict the use to the signal strength and/or the signal-to-noise ratio of the received signal, and to measure the amount of packet losses. A handover is executed only if the new AP seems to provide better connection quality over some stability/dwell period, avoiding a "ping-pong" effect when the MT crosses the overlapping edges of two cells and the signal from the connected AP usually shows significant fluctuations. Considering that each AP switch involves time-consuming *authentication* and *reassociation* procedures (e.g., see [9]), it is important to avoid unnecessary handovers by carefully assigning the values of all the used parameters and thresholds.

Also in the case of vertical handover from WLAN to WWAN and vice versa, a threshold-based scheme (sometimes combined with additional adaptive or fuzzy mechanisms) has been adopted in the majority of the studies(e.g, [10], [11], [12], [13], [14]). Using dual-mode or multiple NICs, it is possible to monitor alternative connection points while using the current network connection (even if this has a negative impact of the on-board energy consumption). Due to the bandwidth and coverage differences existing between WLANs and WWANs, as rule of thumb most of the decision schemes tend to always favor WLAN to WWAN handovers (*upward link* handovers) in case of high speed, while tend to favor WWAN to WLAN handovers (*backward link* handovers) for low speeds. Often, load/bandwidth information is not even took into account, relying on the fact that WLAN's bandwidth outperforms WWAN's bandwidth even under quite high loads such that it is always preferable to switch to a WLAN (see [12] for an argument against this strategy). This approach is also motivated by the intrinsic difficulty for an MT to derive sound estimates of the *available bandwidth* (see also [15], [16] for related work in wired and wireless environments). On the other hand, for the connection points it is relatively easy to measure it, such that they could provide this information to help the MT to take its decisions (realizing a mobile executed but *network assisted handovers*). Unfortunately, given the current status and practical implementations of the IEEE 802.11 standards, in current WLANs MTs cannot rely on *any* feedback from the APs. Indeed, this fact puts strong limitations on the actual optimization of mobile executed handovers (the activities of the IEEE 802.11k working group are precisely addressed to remove these limitations in the deployed standards). However, the work is still in progress and it is still unclear how soon and if the new specifications will be formally adopted (see [17] for a discussion about problems related to disclosing user/network information in both network and mobile assisted handovers).

Using the same terminology adopted in [13], the whole handover process can be practically decomposed in to three functional blocks:

- *Handover Initiation*
- *Network Selection*
- *Handover Execution*

Handover Initiation consists of the *proactive monitoring* of the current connection and/or of possible alternative connections in order to: (i) effectively *anticipate* or explicitly deal with imperative handovers, or (ii) trigger alternative handovers in order to *optimize* costs and performance. In our case, the CNAPT *Search* and *Check Activities* (see Section III-B) both participate to the Handover Initiation process, which is however mostly focused on the treatment of imperative handovers. Network Selection comprises the procedures to select the new connection point according to *decision metrics* like quality of the signal, cost, bandwidth. etc.. Information about these metrics can be gathered either proactively and/or reactively according to the proposed scheme and to the limitations imposed by the used protocols and technology. In our case, Network Selection is supported by the results provided by the Search Activity. Handover Execution stands for the set of procedures to be carried out for the authentication and reassociation of the MT. This pertains to the WiOptiMo CNAPT/SNAPT switching procedure (see Section III-A).

Assessing the goodness of the current and alternative network connections requires, for both Handover Initiation and Network Selection, the proactive and/or reactive passive gathering of PHY data concerning the behavior of signal quality and/or MAC/TCP/APP layer data about the effective bandwidth and latency associated to the wireless link (e.g., [10], [18]). More complex approaches consider also mobility/location information [19], or learning user preferences and behavior [20]. The use of active monitoring based on probing packets to estimate load conditions has also received some attention [16].

## III. The WiOptiMo System

WiOptiMo [1], [2], [21] is a solution for seamless handover among heterogeneous networks/providers. That is, it transparently provides persistent connectivity to users moving across different wired and wireless networks. WiOptiMo detects the available network access points and provides, in automatic or semi-automatic/assisted way, the best Internet connection in terms of estimated QoS (e.g., bandwidth, reliability, and security) and/or cost effectiveness among all the available connections at a certain time and location. The optimized handover is executed without interrupting active network applications or sessions and avoiding or minimizing user intervention. Furthermore, if the current connection becomes no longer available and if no other connections can be established (e.g., inside an uncovered area), the system hibernates the applications to perform re-establishment when the current or a new connection becomes available again (obviously, if the reestablishment exceeds the application timeout, the application may detect a network problem).

In the following subsections we briefly describe the main characteristics of the original WiOptiMo system. From Section IV onward, we discuss the rationale behind the design of the new version of WiOptiMo, and we present the already implemented modifications as well as the general characteristics of the modifications that are still under implementation and testing.

### *A. WiOptiMo Application Layer Solution*

The WiOptiMo system does not require any modification of the layers of the OSI protocol stack and it does not introduce any additional sub-layer. The seamless handover is obtained by a pair of applications (OSI Layer 7), the *CNAPT* (Client Network Address and Port Translator) and the *SNAPT* (Server Network Address and Port Translator), which deceive the communicating Client and the Server applications letting them believe that they are running on the same device, or on different devices belonging to the same network. The Client and the Server applications do not realize that they are communicating via the Internet. The CNAPT and the SNAPT collectively act as a *middleware* and interface the communication between the Client and the Server applications hiding the mobility to them. The CNAPT is an application that can be installed in the same device as the Client application or in a different device in the same mobile network (e.g., in the case of a team of consultants or auditors that require mobility while working together, the CNAPT can be installed in only one of the mobile devices of the mobile network and the whole team can share the seamless handover provided by it). The SNAPT is an application that can be installed in the same device as the Server application or in a different device of the same network or in any Internet server (e.g., in a corporate front-end server, in the home PC, or in any Internet node or router). Thanks to this flexibility, the mobility of multiple users can be handled either using a star topology, with central servers with large computational capabilities and large bandwidth and managed by telecommunication companies or ISP, or using a distributed topology, in which every user manages its mobility by installing the SNAPT on the accessible nodes (e.g., in the home PC if directly connected to the Internet), saving in terms of transmission costs. With the distributed topology, WiOptiMo can provide a sort of ”democratic” seamless handover.

### *B. WiOptiMo Handover Initiation and Network Selection:* Search *and* Check *Activities*

The CNAPT application acts as an application relay system, and also activates a decision task in order to provide persistent and optimized Internet connectivity. The decision task consists of two main activities: the *Search Activity*, for soft handovers, which proactively searches for new network providers and connectivity, and the *Check Activity*, for hard handovers, which continuously monitors reliability and performance of the current connection. Moreover, at any time, the user can manually ask the decision task to switch to another available network connection. This can be useful when the user wants to use a specific network that would be not selected otherwise.

*1) Search Activity:* The Search Activity periodically searches for other available network connections. In the forthcoming new version of the system, this will be done without disturbing the current connection, even in the case that both the current and the checked connection are WLAN (the new behavior builds on the results reported in [22]). The handover to a new network can be triggered in either manual or automatic mode. In *manual mode*, when the Search Activity finds at least one network that could provide an Internet connection better than the current one (based on the parameters input by the user), it asks the user whether she/he wants to switch. In *automatic mode*, the Search Activity autonomously decides whether to execute or not the handover (again, the decision is made on the basis of a set of parameters assigned by the user). In some cases, in automatic mode, the Search Activity might still require some minimal user interaction to the complete the network association procedures. In all cases, from the running application point of view, the Search Activity avoids any interruption of the service during the handover. After the handover has been performed, the user can choose to keep the old Internet connection, otherwise it will be closed in order to reduce power consumption.

*2) Check Activity:* Following a periodic activation scheme, the Check Activity verifies reliability and performance of the current connection. In the current implementation the check interval is a parameter set to one second. If the reliability or performance index go below some specified critical thresholds (possibly set by the user), or the current network connection is experiencing an interruption, the Check Activity tries to switch to a new network provider, or tries to set up a new Internet connection from the same provider if the signal comes back (i.e., the disconnection was only a temporary problem, like when crossing a small uncovered area during a UMTS connection). If the Check Activity does not find any available network providers, it notifies the user that the current connection will be no longer available and changes its operational mode to *Tunneling mode*, which consists of a continuous search for an available network connection. When it eventually finds at least one available connection point, it automatically establishes the connection and comes back to its normal operational mode. During the switch, the Check Activity avoids any interruption of the service. After the switch has been performed, the user can still choose to close the old Internet connection, if it is still alive, in order to reduce power consumption.

## IV. Cross-Layering and Autonomic Design

Given the complexity and the multiple dynamic aspects of internetwork roaming, a design based on a *cross-layering* architecture [5] interleaved with *autonomic* components [3] is almost an unescapable choice to achieve adaptive behavior and performance optimization. As it is also witnessed by other approaches to handover optimization (e.g., [10], [23], [6]), cross-layer data are necessary to collect the information necessary to deal with the intrinsic variability and uncertainty associated to the physical measures of interest like the Received Signal Strength (RSS) and the available bandwidth. On the other hand, autonomic components are based on the proactive monitoring of the system's performance and behavior which, in turn, results in the self-optimization of internal thresholds and in the continual adaptation of the delivered Quality of Service (QoS) to the ever changing external conditions. Adopting this approach, the user can experience seamless connectivity in a fully transparent way, letting the system to adapt to his/her needs and to struggle to provide the required QoS.

The general architecture of the new WiOptiMo system consists of three main functional components: (i) *cross-layer monitoring*, that performs active and passive monitoring activities and data collection at the physical, network, and application layers, (ii) *self-optimization and learning*, carried out by an *optimization module* which makes use of a repository of past experiences to adapt internal parameters and derive statistical measures of trend, (iii) *interaction with the user*, intended to get user's feedback for the selected handovers and to allow the user to input its preferences' profile through an intuitive interface.

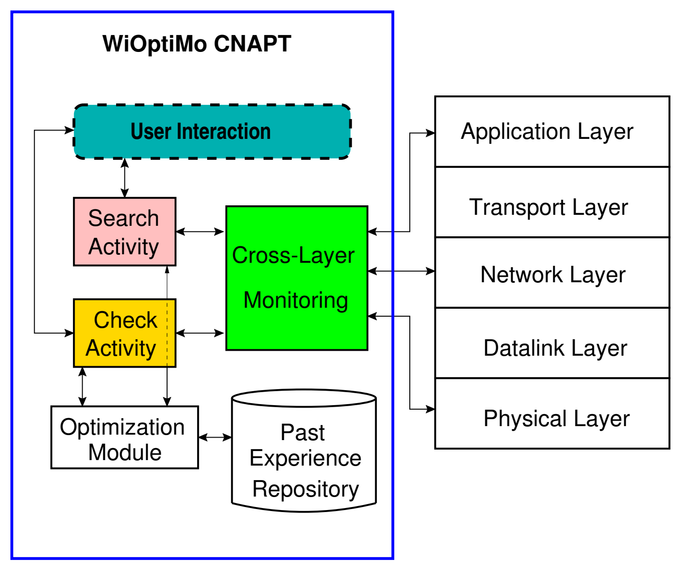


Fig. 1. Functional components in the WiOptiMo CNAPT module

Both the original Check Activity and Search Activity components have been re-designed according to this combination of cross-layering and autonomic approach. In the following we provide a general discussion on cross-layering and autonomic issues in WiOptiMo, as well as a description of their implementation in the new version of WiOptiMo. However, since the new release of the system is still undergoing full development and testing, we keep the discussion at a rather general level for those aspects that have not been fully implemented and carefully tested.

### A. Cross-Layer Monitoring of Network Connections

Simple PHY monitoring, while necessary to understand the physical status of the connection, is unable to offer information on the connection traffic load. On the other hand, we could not take into account LINK monitoring of variables related to frame count, since apparently some NICs do not allow to read these values using standard APIs. Therefore, in our novel design, we integrated the application layer approach of WiOptiMo system with the access to information and protocols belonging to other layers, and in particular to the physical layer. From one side, this solution introduces interdependencies of the WiOptiMo system with other layers, which create additional complexity at design and implementation time. On the other side, we observed in practice that this solution can really allow to optimize system performance without having a negative impact on the overall efficiency of the system.

*1) Physical Layer Monitoring:* ¿From our experiments, we derived that physical monitoring is significant and reliable only for few parameters, namely the RSS and FCS, and we decided to use the RSS, as the FCS does not offer more information than the RSS [21], [24]. In the original WiOptiMo system, we decided to perform periodic (every second) sampling of the RSS. The RSS value is used raw, without further smoothing. The use of raw values and such a low sampling rate was justified by the need of keeping as low as possible the computational load on the mobile device. However, due to the high and frequent fluctuations of the raw RSS, we realized that we need richer and better information to derive proper conclusions on the evolution of the RSS. At this aim, in the new version the system will read RSS information for each received frame. In this way, it will have sufficient data to be able to robustly smooth the sampled values through simple weighted moving averages, and at the same time calculate a simple trend indicator to be used in cross-validation with the moving average (both the chosen statistical indicators can be efficiently implemented using elementary integer calculations).

*2) Application Layer Monitoring:* WiOptiMo conveys all the traffic exchanged between the client and the server applications on the CNAPT/SNAPT connection, and the CNAPT knows at each time whether there is or not some traffic. Therefore the system can easily monitor the connection at this layer, perform further check on the quality of the connection, and derive useful information. While it is not feasible to interact with the information contained in all the exchanged data packets, as it would be too expensive, we found appropriate to carry out active measurements at this layer by injecting few control packets in the connection and observe their behavior. More specifically, we send *ping messages* (ICMP packets) to the access point, for the purpose of:

- Estimate the connection in terms of available throughput,on the basis of the experienced Round-Trip-Time (RTT). This gives some indication on:
  1) the *effective status* of the connection (if it is active or not), as the status detected at the physical layer could be not correct, as discussed in Section VI
  2) the *effective load* of the connection, which would not be possible to infer from the measures available at physical layer. We are aware that ICMP traffic is controll traffic. However, as we are not trying to quantitatively measure the throughput, but we simply want to have a dynamic estimation of the WLAN charge, this is sufficient for our purpose.
- Avoid to improperly react to temporary fluctuations of physical parameters. While a negative indication from physical monitoring (e.g., a very low RSS level) corresponds to a high probability that the physical connection (and consequently the CNAPT/SNAPT connection) is not anymore valid, this can be due just to some temporary cause. On the other hand, if a short train of ping messages experience a timeout, this evidence can provide a more robust confirmation of the fact that a link is down.

### B. Autonomic Design

In previous work [21], [24] we conducted extensive experimental tests in real-world scenarios to gain insights on the possible metrics and strategies that can be adopted to optimize the procedures of internetwork roaming. One of the results that has emerged from our experiments is that *the goodness of a metric or of a parameter value, strongly depends on the wireless scenario at hand*. For instance, the behavior of the RSS can be a good or bad predictor of the quality of the connection depending on the considered scenario and on the specific hardware used. Therefore, this tells us that an optimization mechanism based on static parameters and strategies is not apt to deal satisfactorily with the large variety of wireless scenarios of practical interest. Our approach to solve this issue is to *empower the cross-layering architecture with an autonomic design*: our system is able to *adapt* and *self-optimize/tune* its internal parameters and performance according to an understanding of both the current context (wireless scenario inclusive of the specifically used equipment) and the user preferences and mobility patterns. In order to show a truly adaptive behavior, and in some extent predict and anticipate changing in the environment and in the user activities/mobility, it is necessary to *learn from past experience*. At this aim, the new WiOptiMo will include a repository of the most significant information (trends, failures, trajectories, user choices, etc.) about past experience. For instance, if an on-board GPS is available, the system can decide to store and learn maps identifying good coverage areas together with the characteristics of the network access that can provide the coverage. This might result quite useful (and relatively easy to learn) in the case of users constantly travel along the same routes, as it is the case for people daily commuting between their homes and work places.

The WiOptiMo system is intended to run unchanged on the majority of the portable devices in commerce. This is reflected in the fact that the system can self-detect the characteristics of the on-board hardware and take the appropriate action flows (see Section VI). This is a fundamental *self-configuration* feature already present in the system, that, together with the discussed properties of self-optimization and and self-tuning, support even more the view of WiOptiMo modules as autonomic modules.

*1) A concrete example: Self-tuning of the RSS threshold:* The default RSS threshold value that we used in the original WiOptiMo system to check connection reliability (in both the Check and Search activities), does not always reflect the characteristics of the current context in terms of signal characteristics at the specific location, and hardware and software configuration. In order to achieve adaptive context-dependent tuning, we designed a simple adaptive component for the self-tuning of the RSS threshold according to the current context. We continuously check if the system, when it looses the connection (i.e., an IP address is not anymore available), has reached an RSS value lower than that of the currently stored RSS threshold value. If so, we assume that this new lower value is the inferior limit for successful communications at the current location, given the hardware and software configuration. Therefore, we take this new value to adapt the value of the variable containing the RSS threshold, and we start using it. This behavior can be started either by the system, under reception of an event of loss connection, or by the users, for example when she/he installs a new hardware. The current implementation, which can be only started manually, requires that the user moves around for a while, in order to gather the required information. However, in the final release of the next version we intend to fully automate the procedure.

### *C. User Interaction*

An important part in the definition of the context that the optimization strategy has to take into account, is played by the user itself, with its current preferences. In many cases, in practice it is not possible to assume that the system can infer/learn user's preferences in relationship to its needs at the current time and location. In order to provide a satisfactory service some user interaction must be assumed. For instance, if a specific download is very urgent, bandwidth might be more important than cost, but this cannot be known in advance to the system without additional information from the user, who has to input is current QoS requirements in its *user profile*. Moreover, some additional information concerning for instance the "typical" current speed of the user (e.g., vehicular, pedestrian, etc.) might be of great help to the system, even if this information might be inferred from lower layer data. At this aim, we are realizing an intuitive and user-friendly user interface, to facilitate the input and the updating of profiling information. We are also considering mechanisms to trigger requests of user-assistance in case of very ambiguous situations.

## V. Experimental Results for Measures of Load Estimation at the Application Layer

Experimental results concerning work in the domain of vertical handover optimization are mainly restricted to simulations (a notable exception is reported in [25]). An important negative consequence of this consists in the fact that it is really hard to assess the validity of the proposed simulation-based approaches in the perspective of the implementation in real-world networks [26]. On the other hand, as already pointed out, our choice is to face the real-world challenges. In this perspective, and we the aim of gathering experience for a first validation for the use of cross-layering monitoring, we realized a set of real-world experiments focused on the use of ping messages (see IV-A.2) for the establishment of the effective status of the connection, and for channel load estimation.

The soundness and performance of the (original) WiOptiMo system in the case of using only simple physical monitoring was assessed in previous work [1]. In [21] we reported extensive results concerning the effectiveness of a number of different metrics at the physical and MAC layers. Here we present a set of experiments aimed at assessing the utility and the usability of integrating physical layer controls with application layer controls. The results of these experiments gave us a further motivation to proceed along the way of integrating cross-layer monitoring and control in the WiOptiMo system, as it is discussed in the next section.

All the experiments were conducted in favorable conditions of strong and stable RSS, to see if, given good physical conditions for the connection, we could observe different indications at the application layer in relationship to variations in the wireless scenario. More specifically, we report the results concerning the use of ping messages to derive load estimates at the application layer as a function of variations in the load offered to the wireless environment.

Ping messages are proactively sent to the access point in order to evaluate the connection status in terms of: (i) *throughput*, on the basis of the observed RTTs, and (ii) *channel congestion* according to the number of experienced timeouts. To understand whether or not the RTT of ping messages can be used as a robust indicator of channel congestion, we performed traffic-intensive experiments with an increasing load offered to the WLAN. We considered the following four scenarios: (1) an increasing number of laptops (from 1 to 10) continuously download data from one PC connected via LAN to the AP, while another laptop sends the ping messages to the AP; (2) an increasing number of laptops (from 1 to 7) continuously download data from each other passing through the AP, while another laptop sends ping messages to the AP (in this way the network load was doubled with respect to the previous case); (3) an increasing number of laptops (from 1 to 10) continuously download data from the Internet through the AP, while another laptop sends the ping messages to the AP; (4) 11 laptops with an increasing number of downloading processes (from 1 to 5 on each laptop) from the Internet through the AP, while another laptop sends the ping messages to the AP. In each scenario a continuous train of 1000 ping messages (size of 32 bytes) was generated during the experiment. The size of each single download was of 1.5 Gbytes. The WLAN is based on 802.11g devices.

As shown in Figures 2 and 3, in the first two scenarios the average RTT and the number of experienced timeouts increased each time that a new laptop joined the network with a new download. Moreover, the number of timeout rose up very fast when the number of laptops in download grew from 7 to 10 in the first scenario, and from 3 to 7 in the second scenario.

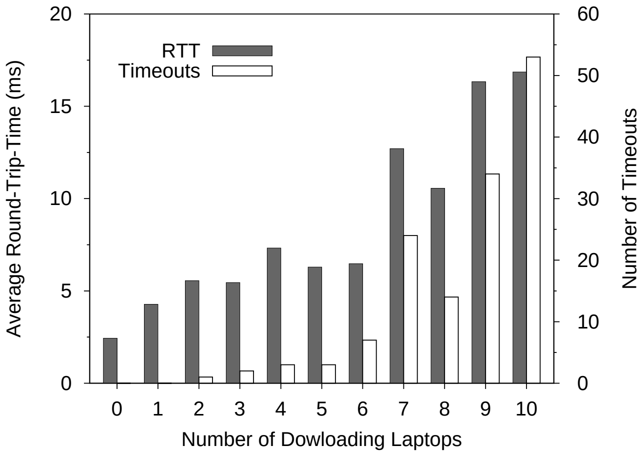


Fig. 2. Experimental results for the first load estimation scenario. An increasing number of laptops download data from one PC connected via LAN to the AP while another laptop sends ping messages to the AP.

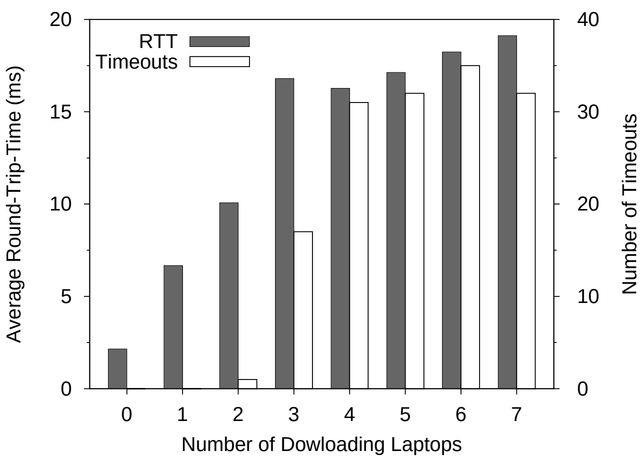


Fig. 3. Experimental results for the second load estimation scenario. An increasing number of laptops download data from each other passing through the AP while another laptop sends ping messages to the AP.

In these scenarios, the RSS was always above the critical threshold. Therefore, no handover indications could have been derived from this physical measure. However, from Figures 2 and 3, it is clear that the throughput was significantly decreasing, and the channel congestion increasing, with the increase of the active users in the WLAN. Therefore, the combined analysis of the RSS and RTT values seems to be necessary to become aware of a situation of pure traffic congestion.

On the contrary, in the third scenario (Figure 4) the both the average RTT and the number of timeouts remain low almost independently from the number of laptops in download. This was probably due to the available bandwidth on the Internet side, that was lower than that available in the WLAN. In the last scenario we overloaded the WLAN with a large number of downloads (11 up to 53) from the Internet. As shown in Figure 5, the average RTT remained more or less the same but the number of timeouts rose up very fast reaching more than 10% of the total number of ping messages sent in the worst case. Therefore, in these last two scenarios, the advantage of a cross-layering approach have been less evident, but still useful to estimate the quality of the connection throughput.

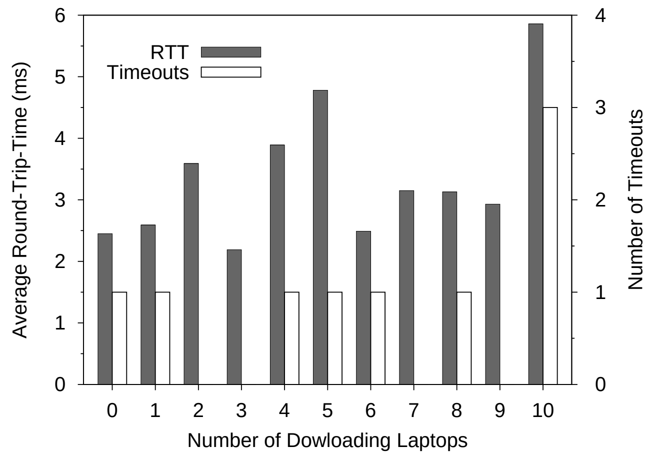


Fig. 4. Experimental results for the third load estimation scenario. An increasing number of laptops download data from the Internet through the AP while another laptop sends ping messages to the AP.

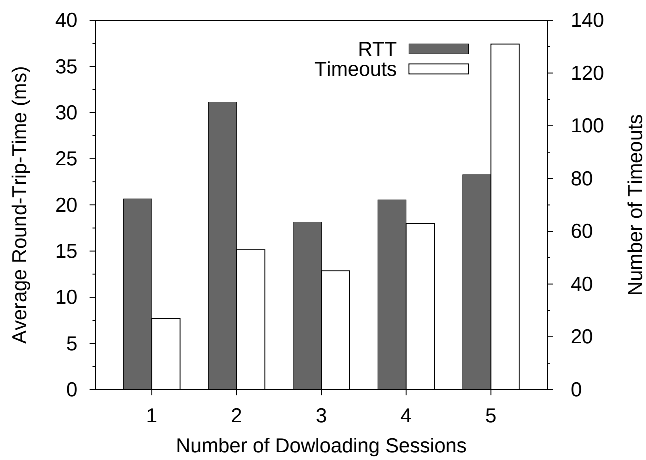


Fig. 5. Experimental results for the fourth load estimation scenario. 11 laptops with an increasing number of sessions download data from the Internet through the AP while another laptop sends ping messages to the AP.

## VI. The New Check Activity

The experimental results discussed in the previous section seem to indicate that the information obtainable at the application layer can fruitfully complement the information from the physical layer to derive more robust estimations of the status of a connection link. As shown by the previous experiments, we can have discordant indications between physical and application layer monitoring of the connection, such that they should be used in cross-validation. This is one of the main reasons behind our choice to adopt a cross-layer approach and modify the Search and Check Activity according to the design ideas presented in IV. Here we present the basic structure of the new Check Activity, which includes both *passive monitoring at the physical layer* and *active monitoring at the network/application layer*. In the new Check Activity, the use of ping packets, even if not fully exploited, resulted very useful, as discussed below. The new Check Activity repeatedly interacts with the user, and tries to learn from past experience, to adapt and self-tune internal parameters and derive trend measures (for the RSS). However, we are still testing and improving these mechanisms, such that they are not further discussed in the following.

The flowchart of the actions of the new Check Activity based on cross-layering design ideas is reported in Figure 6. The flowchart shows only the stable core of the function, omitting the parts still under developments.

The Check Activity first checks the presence of the IP address for the current connection (*Network Layer check*). If the IP address is present (i.e., from the operating system point of view, the communication device miniport is connected), it verifies that the current connection is a wired connection, (e.g., Ethernet LAN, ADSL, Token ring, FDDI). This phase is called *First Physical check*. If this is the case, no additional checks are needed, as the wired connections are considered as always good and reliable connections. Otherwise, if the connection is a wireless one and it is in use that is, there is at least one application or service that needs to access the Internet (*First Application Layer check*), the Check Activity performs a number of additional actions which depend on the type of devices used to make the wireless connections. These devices are grouped in three categories:

1) Wireless WAN Programmable devices: GPRS, EDGE, UMTS, HSDPA, CDMA 1x or EV-DO PCMCIA, and USB or CF devices providing an API SDK (e.g., Nokia D211/311 and Sierra Wireless PC cards), that gives the possibility to establish/destroy the wireless connection and control all its parameters;
2) Wi-Fi devices;
3) Wireless WAN Dial-Up Networking (DUN) modems: GPRS, EDGE, UMTS, HSDPA, CDMA 1x or EV-DO PCMCIA, USB, CF devices or mobile phones accessible via USB, Serial Cable or Bluetooth that can be controlled only via standard AT commands.

In general, for the third category, once the DUN connection has been established, it is not anymore possible to access the connection parameters (e.g., the signal strength), as the COM port used for the interaction is held by the operating system to provide the WAN connection. [2] Therefore, the Check Activity cannot perform any further physical check. On the contrary, for the Wireless WAN Programmable devices and for the Wi-Fi devices, a check on the signal strength is performed (*Second Physical Layer check*). If the RSS is greater than a *critical RSS threshold*, or if a Wireless WAN Dial-Up Networking modem is used, the Check Activity controls the network activity for the active applications or services (*Second Application Layer check*). If an exchange of data is going

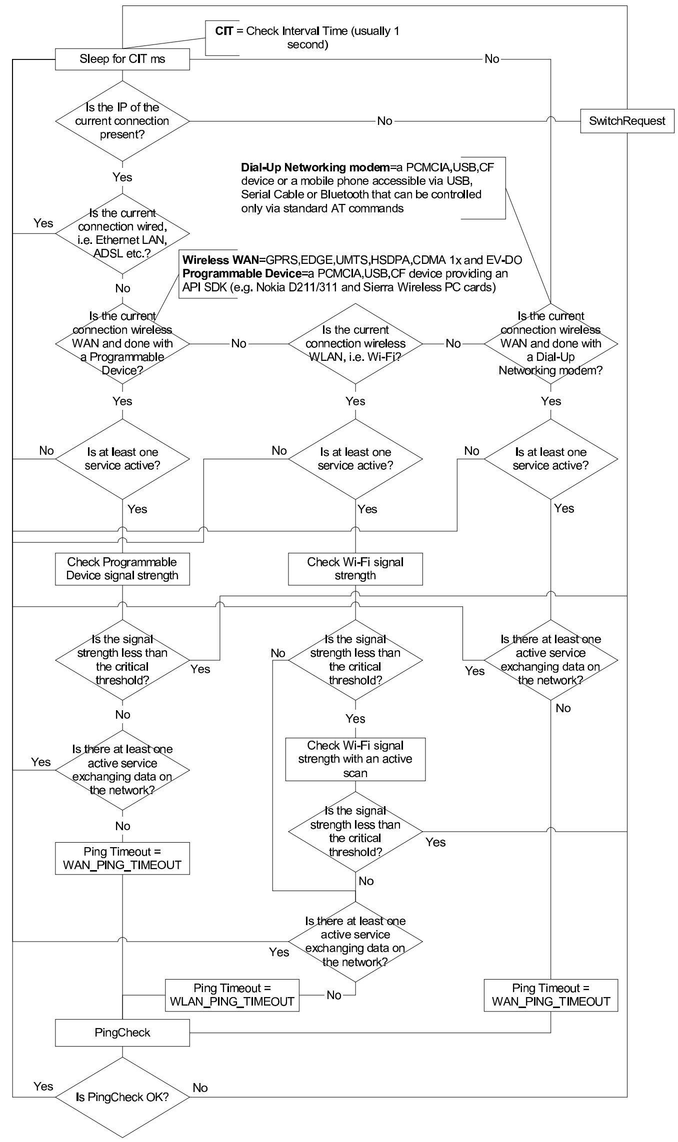


Fig. 6. Flowchart of the Check Activity function

[2] As already discussed, WiOptiMo is intended to run unchanged on the majority of the portable devices in commerce, thus it has a modular architecture, with the core module being platform-independent (JAVA). The other modules are platform-dependent. Considered that as a matter of fact, the majority of portable devices rely on the use of some version of Windows as operating system, the other modules has been developed in Windows environment, assuming the Windows' NDIS driver for the communication with the NIC. Therefore, in the following all the references to the operating system and related issues have to be thought as referred to Windows.

on, the current connection is considered *good*. Otherwise, a *ping check* is started (*Third Application Layer check*). The ping check sends three ICMP ECHO REQUEST packets to a reachable host (or if it is not possible, it establishes three TCP connections with a reachable host), and starts a timeout timer. If all the three attempts are not echoed within the requested timeout, the connection is considered lost and a handover is requested. Actually, since it is expensive in terms of time and cost (e.g., in case of GPRS connections), the ping check is not performed at every single run of the Check Activity, but with a frequency modulated by the number of timeouts observed during previous runs. This simple but rather effective strategy is similar to that adopted in [27], where the authors show some empirical evidence that the fall of a connection can be robustly assessed after three consecutive frame losses (without considering any additional metric).

We performed some tests with this new implementation and observed that the new design is very beneficial for:

1) DUN connections. Cross-layer monitoring allows to bypass the impossibility of physical monitoring, as explained above, and is the only way to check if an IP connection is still alive or not.
2) WLAN connections. If the RSS suddenly drops, some cards indicate $-200dBm$, while some other cards keep on reporting the same RSS value as before the drop or even other values. For this reason, the reported RSS value is not always a reliable indicator of good signal. On the other hand, cross-layer monitoring provides additional information on the status of the connection, and allows to understand in reliable way if it is still alive or not.

According to these facts and experimental evidences, we are currently studying a lightweight and robust solution to use ping packets for deriving load estimates.

## VII. Conclusions

In line with the ubiquitous computing vision, mobile devices are becoming part of our daily life. In the majority of the real-world scenarios, due to topology, location, and time constraints, the only way to provide persistent communications and satisfactory QoS performance, is to switch from one provider to another, always trying to stay connected using the best possible connection in terms of bandwidth, cost, and coverage. However, due to the lack of systems that can really guarantee optimized seamless handovers, this revolution has not yet become real. Several solutions have been proposed in literature, but they are hardly implementable and usable in real systems. In this paper we introduced innovative design solutions to boost-up performance and flexibility of WiOptiMo, our system for internetwork roaming.

Up to our knowledge, WiOptiMo is the first solution able to work in general real-world scenarios in terms of used network cards, providers, applications, etc. We presented an innovative cross-layering and autonomic design for the forthcoming next release of WiOptiMo, and we performed some real-world experiments to support the choice of the novel design. Considering several scenarios, we found that cross-layering monitoring and control can be very beneficial, as it can allow to discover situations requiring handovers that otherwise would go unnoticed. We also point out that we made not specific assumptions, and that we are working for a system that can work in any context.

Given the preliminary encouraging results, which validated our approach, we are designing new components for quantitative analysis that will enhance the autonomy of the system and the interaction with the user.

In our current work, we are including statistical evaluations for the results, the use of different type of traffic in the wireless network (to ensure the generality of our solution with respect to the traffic generated) and the monitoring of other parameters to further improve the response of the system. Furthermore, we are also investigating the use of techniques for efficiently smoothing the noisy RSS and for the calculation of trend measures to be used in cross-validation with the RSS to add robustness to Handover Initiation decisions. Moreover, we will explore the use of learning techniques based on the use of past experience to improve the overall autonomy of the system. This work is expect to further improve the already very satisfactory performance and flexibility of the system.

## References


[1] S. Giordano, D. Lenzarini, A. Puiatti, and S. Vanini, "WiSwitch: seamleass handover between multi-provider networks," in *Proceedings of the 2nd Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2005.

[2] S. Giordano, D. Lenzarini, M. Schiavoni, and S. Vanin, "Virtual web channel: Flow aggregation for enhanced ubiquitous web access," in *Proceedings of IEEE WirelessCom*, 2005.

[3] J. Kephart and D. Chess, "The vision of autonomic computing," *IEEE Computer magazine*, pp. 41–50, January 2003.

[4] D. F. Bantz, C. Bisdikian, D. Challener, J. P. Karidis, S. Mastrianni, A. Mohindra, D. G. Shea, and M. Vanover, "Autonomic personal computing," *IBM Systems Journal*, vol. 42, no. 1, pp. 165–176, 2003.

[5] M. Conti, S. Giordano, G. Maselli, and G. Turi, "Cross-layering in mobile ad hoc network design," *IEEE Computer*, February 2004.

[6] S. Shakkottai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks," in *Proceedings of IEEE Communications*, 2003.

[7] N. Tripathi, J. Reed, and H. Vanlandingham, "Handoff in cellular systems," *IEEE Personal Communications Magazine*, December 1998.

[8] M.-H. Ye, Y. Liu, and H. Zhang, "The mobile IP handoff between hybrid networks," in *Proceedings of the 12th Annual IEEE Int. Symposium on Personal Indoor and Mobile Radio Communications*, 2002.

[9] S. Pack, H. Jung, T. Kwon, and Y. Choi, "SNC: A selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks," *ACM Mobile Computing and Communications Review*, vol. June, 2005.

[10] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *Journal of Selected Areas in Communications* Special Issue on Advanced Mobility Management And QoS Protocols for Next-generation Wireless Internet, vol. 22, no. 5, June 2004.

[11] R. Inayat, R. Aibara, and K. Nishimura, "Providing seamless communications through heterogeneous wireless IP networks," in *Proceedings of International Conference on Wireless Networks*, 2004.

[12] M. Ylianttila, M. Pande, J. Mäkelä, and P. Mähönen, "Optimization scheme for mobile users performing vertical handoffs between IEEE 802.11 and GPRS/EDGE networks," in *Proceedings of Global Telecommunications Conference 2001*, 2001.

[13] K. Murray and D. Pesch, "State of the art: Admission control and mobility management in heterogeneous wireless networks," Deliverable 3.1 D1.1 of the *M-Zones* Project, May 2003.

[14] A. Majlesi and B. Khalaj, "An adaptive fuzzy logic based handoff algorithm for hybrid networks," in *Proceedings of the International Conference on Signal Processing*, August 2002.
[15] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas on Communications* Special Issue on Internet and WWW Measurement, Mapping, and Modeling, vol. 21, no. 6, pp. 879–894, 2003.
[16] A. Johnsson, "Bandwidth measurements in wired and wireless networks," Ph.D. dissertation, Mälardalen Research and Technology Centre (MRTC), Mälardalen University, Mälardalen, Sweden, April 2005.
[17] W. Zhang, J. Jaehnert, and K. Dolzer, "Design and evaluation of a handover decision strategy for 4th generation mobile networks," in *Proceedings of the 57th Semiannual Vehicular Tech. Conf. (VTC)*, 2003.
[18] B. D. Noble and M. Satyanarayanan, "Experience with adaptive mobile applications in Odyssey," *Mobile Networks and Appls.*, vol. 4, 1999.
[19] A. Saleh, "A location-aided decision algorithm for handoff across heterogeneous wireless overlay networks," Master's thesis, Virginia Polytechnic Institute and State University, 2004.
[20] G. Lee, P. Faratin, S. Bauer, and J. Wroclawski, "A user-guided cognitive agent for network service selection in pervasive computing environments," in *Proceedings of 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2004.
[21] S. Giordano, M. Kulig, D. Lenzarini, A. Puiatti, F. Schwitter, and S. Vanin, "WiOptiMo: Optimised seamless handover," in *Proceedings of IEEE WPMC*, 2005.
[22] R. Chandra, P. Bahl, and V. Bahl, "MultiNet: connecting to multiple IEEE 802.11 networks using a single woreless card," in *Proceedings of IEEE INFOCOM*, 2004.
[23] K. Byoung-Jo, "A network service for providing wireless channel information for adaptive mobile applications: Proposal," in *Proceedings of the ICC01*, vol. 5, 2001, pp. 1345–1351.
[24] M. Kulig and F. Schwitter, "Monitoraggio dei parametri per ottimizzare una soluzione di always-on," in *Semester Project Report - SUPSI*, 2005.
[25] P. Vidales, G. Mapp, F. Stajano, and J. Crowcroft, "A practical approach for 4G systems: Deployment of overlay networks," in *Proceedings of Tridentcom*, 2005.
[26] D. Kotz, C. Newport, R. Gray, Y. Yuan, and C. Elliot, "Experimental evaluation of wireless simulation assumptions," 2004.
[27] H. Velayos and G. Karlsson, "Techniques to reduce ieee 802.11b mac layer handover time," Kungliga Tekniska Hgskolan (KTH), Stockholm, Sweden, Tech. Rep. TRITA-IMIT-LCN R 03:02, April 2003.