Always Best Packet Switching: the Mobile VoIP Case Study

Vittorio Ghini, Giorgia Lodi and Fabio Panzieri
Dept. of Computer Science, University of Bologna, Italy
Email: {ghini, lodig, panzieri}@cs.unibo.it

Abstract - **Currently, a mobile device equipped with multiple heterogeneous wireless Network Interface Cards (NICs), cannot take fully advantage of its capabilities as the conventional mobility management approach allows it to select and use only one of its NICs at a time. In contrast to this approach, we propose the so-called Always Best Packet Switching (ABPS) model that allows applications to use simultaneously all the available NICs, and switch each given IP datagram through the most suitable NIC, depending on the characteristics of the datagram itself. Specifically, we show that the ABPS model enables mobile applications to create policies for load balancing and recovery purposes in order to support effectively mobile multimedia services.**

In particular, in this paper we describe a cross-layer mechanism, termed Robust Wireless Multi-Path Channel (RWMPC) that adopts the ABPS model in order to meet effectively interactivity and low packet loss Quality of Service (QoS) requirements of Voice over IP (VoIP) applications, running on mobile hosts equipped with multiple WLAN NICs. By Exploiting the RWMPC services, VoIP applications can establish and maintain separate wireless links with different access points providing access to, possibly independently managed, wireless networks, and select for each UDP datagram the best link to be used for communication purposes. In essence, the RWMPC mechanism monitors each link in use, and notifies the application if detects that a datagram has been lost. In this case, the application retransmits the datagram using an alternative link, available through one of the interfaces homed in the mobile host.

We have carried out an experimental evaluation of our mechanism in a real wireless scenario through the emulation of a VoIP application. The results we have obtained from this evaluation are discussed in this paper and confirm the effectiveness of our approach.

Index Terms – **Voice over IP (VoIP), Quality of Service (QoS), WiFi, cross-layer protocols, multi-homing, retransmission, Always Best Packet Switching (ABPS).**

I. Introduction

At present, the mobile communications scenario is evolving into two main trends: i) the creation of new wireless broadband communication technologies dedicated to different types of contexts (e.g. Mobile WiMAX, ZigBee), and ii) the growth of broadband WiFi (IEEE802.11a/b/g/n) wireless coverage provided by both access providers (e.g. Boing [1]) and organizations based on cooperation among users (e.g. FONERA [2]).

A mobile device equipped with multiple heterogeneous wireless Network Interface Cards (**NICs**), the so-called **multi-homed Mobile Node** (**MN**), could use the most appropriate technology for each context. Unfortunately, despite the numerous efforts made in recent years, a number of technical, economic and medical-biological limits still prevent mobile users to continuously and effectively use on-line multimedia services such as latency-bound applications based on SIP, UDP and RTP/RTCP protocols or Web-based applications characterized by less strict interactivity requirements.

A. Long-term Objective

Our current researches aim at designing and developing an infrastructure model to support mobile multimedia services, which provide abstractions and seamless communication services, ensuring to the applications on multi-homed MNs uninterrupted availability of communications, high interactivity, limited losses, wide bandwidth, compatibility with existing multimedia applications, high autonomy and low costs. The service abstraction is located at session level, thus masking the complexity of the underlying levels, but it is also designed following a cross-layer approach that involves the lowest levels of the architecture, exploiting the characteristics of the individual technologies. The ability to affect the architecture low-level is implemented on the mobile node by properly modifying the kernel of the most popular open-source operating system GNU/ Linux.

The aim is to build an architecture (shown in Fig. 1) that exploits the multiplicity and heterogeneity of the NICs available on each mobile node, overcoming the problems caused by the handover between different access points, to obtain the following objectives:

1) limit the emission of electromagnetic radiations of mobile devices, thus reducing the exposure and potential risks for the users; this will be obtained choosing at run-time the medium range (i.e., IEEE802.11/b/g/n) communication technologies that, in order to transmit the same quantity of data, radiate energy up to two orders of magnitude less than wide-range technologies (UMTS, WiMAX);

2) improve the availability of bandwidth, both favoring the use of medium-range communication technologies, and using at the same time all available NICs of the MN for communications on parallel tracks;

Manuscript received March 2, 2009; revised May 31, 2009; accepted August 8, 2009.

doi:10.4304/jcm.4.9.700-713

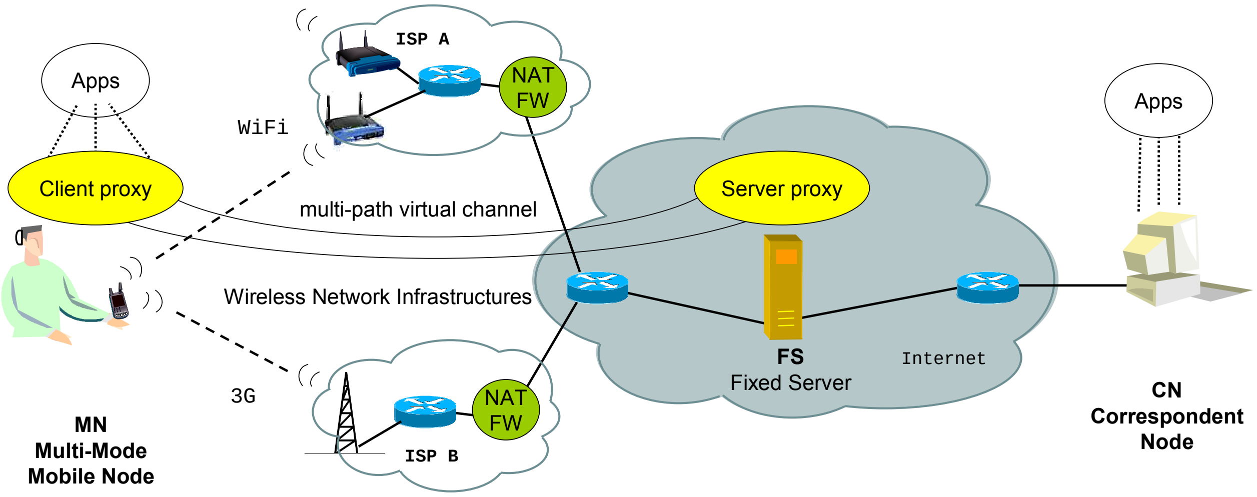

Figure 1. The basic architecture for mobility management and QoS provisioning.

3) provide sufficient end-to-end interactivity, implementing cross-layer protocols for early detection of packet loss and retransmission mechanisms on alternative routes, taking advantage of the multiple NICs of a MN;

4) use existing wireless access infrastructures without introducing changes to them, implementing the necessary protocols on end-systems only, and not relying on the presence, in the access infrastructure, of Mobile IP and IPv6 protocols which are not yet sufficiently diffused;

5) overcome the limitations imposed by the widespread presence of security systems such as Firewalls and NATs, designing the infrastructure and the necessary protocols without resorting to existing fictitious solutions, such as external STUN and TURN systems that would cause performance degradation.

B. *Design Guidelines*

To achieve these objectives, we propose to design the architecture shown in Fig. 1 following two main guidelines, which differentiate our approach from other ones that can be found in literature.

i) The first guideline states that the basic architecture (as depicted in Fig. 1) uses proxy servers that are external to the wireless access networks and not overshadowed by a firewall, and that each MN owns a local communication manager (**client proxy**) that maintains the communication, through the different NICs, with a **server proxy** on the Fixed Server (FS). The server proxy represents the MN in the eyes of the outside world; it is in charge of maintaining the communication continuity with the MN (without using Mobile IP), and integrates the necessary functionalities to overcome the presence of firewalls. Applications on the MN use a multi-path virtual channel between the client proxy and server proxy to communicate with the rest of the world. In this way, the MN may use existing wireless access infrastructures, implementing the necessary mobility protocols only on the two proxies. In other words, the communication continuity is provided as an additional service outside the access infrastructures.

ii) The second and, from the standpoint of this paper, most important guideline refers to an innovative management of the mobility support. Specifically, the conventional approach requires that the MN identifies the best wireless network among those available through the NICs; once identified the MN uses that network as single point of Internet access until the performances degrade. In this case, the MN will select the new best wireless network that will replace the previous one by means of a handover procedure. This is the classic Always Best Connected (ABC) model [3]. In contrast, our approach allows applications to use simultaneously all the available NICs, differentiating the choice of a NIC from datagram to datagram, and introducing the ability to switch to the most suitable path depending on the characteristics of the datagram. We term this model Always Best Packet Switching (**ABPS**): it enables the MN applications to create policies for load balancing and recovery in order to maximize the available bandwidth and decrease the loss rate.

C. *Applying the ABPS model to mobile VoIP*

In this paper we concentrate on the ABPS model only. We do not discuss the overall architecture that, for the scope of this paper, is simply a realistic scenario in which we apply the novel model. Here, we wish to demonstrate that, in a context in which a MN communicates with the Correspondent Node (CN) through a fixed server, the ABPS model may provide a multimedia application executing on the MN with the suitable QoS, despite user movements, packet losses and handovers.

In particular, in this paper we apply the ABPS model to a VoIP application and describe the design, implementation and experimental evaluation of a QoS mechanism, termed **Robust Wireless Multi-Path Channel** (**RWMPC**). The principal objective of this mechanism is to guarantee that QoS requirements of VoIP applications are met according to the ITU-T G.1010 guidelines [4] for interactivity (transmission delay below 150 ms) and low packet loss (below 3% of the transmitted packets).

The design of RWMPC is based on a cross-layer approach through which different information, coming from the diverse ISO/OSI stack abstraction layers, are used by the applications in order to carry out a run time monitoring of the wireless communication links used for voice transmission purposes; in particular, the main differences with respect to others existing approaches can be summarized in the following two points: 1) the monitoring mechanism detects whether each given UDP datagram has been lost during the transmission between a given NIC of the MN and the wireless access point with which the NIC is associated; in addition, it notifies the application that produced the lost datagram; 2) the application may retransmit each lost VoIP UDP datagram using a different NIC.

The basic idea of our monitoring mechanism could be potentially applied to each wireless technology; however, so far we have implemented it only in the IEEE802.11 protocol suite. In particular, we have implemented the cross-layer mechanism in the Linux kernel version 2.6.27.4.

The specific scenario we are considering is illustrated in Fig. 2. We assume there exists a voice communication between two end systems, labeled A and B in Fig. 2. The end system A is a mobile device equipped with two or more wireless NICs, conformant to the IEEE 802.11b/g/n standards [5, 6, 7]. The mobile device A can be located in a metropolitan area where other portable devices exist; these devices use the available wireless network infrastructures for their own transmission purposes. The wireless NICs of the portable devices, including those owned by A, are associated with the APs located in the metropolitan area. The APs are connected to the wired Internet network, to which other users are directly connected, including the end system B.

In this scenario, the entire quality of voice communication is determined by the quality that can be provided by both the wired and wireless environments. However, whereas Internet exhibits a relatively short delay, high bandwidth, and stable links, this is not true for wireless/mobile networks, which are generally characterized by high delays and unstable links (this latter characteristic is principally caused by the handoff procedure of mobile devices which may lose the link and the associated AP as they move). Therefore, as reported by state of the art research works [8], the first hop from the mobile host is the most critical one in the communication, with regard to the level of QoS it provides to VoIP applications.

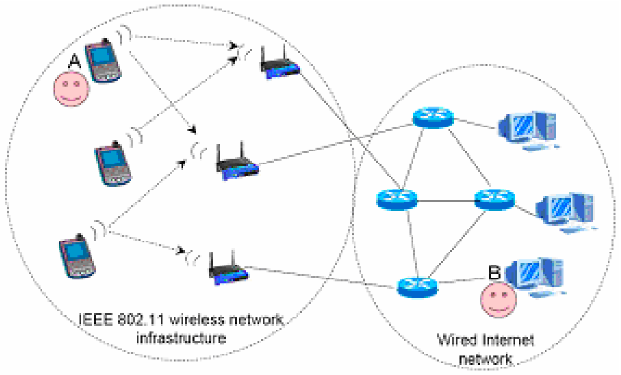

Figure 2. Scenario.

In view of these observations, the RWMPC mechanism we propose is responsible for monitoring the first hop in the wireless communication in order to assess its effectiveness in meeting the interactivity and low packet loss QoS requirements of VoIP applications. As we assume that a mobile host is equipped with multiple wireless NICs, multi-homing capabilities are exploited by our mechanism; that is, our mechanism can establish and maintain separate wireless links with the different APs that provide access to independent wireless networks, and, for each given datagram to be transmitted, it uses one link among those available. This link is monitored by our mechanism in order to detect whether a packet loss occurs. If a packet loss occurs, the RWMPC mechanism retransmits the lost VoIP packet using an alternative link available through one of the NICs homed in the mobile host.

Through an experimental evaluation carried out in a real wireless scenario, available in our city (Bologna), we have been able to confirm the effectiveness of our approach.

Note that the mechanism we propose in this paper is sufficiently general to be applied in a fully wireless/mobile scenario in which both the end systems of the VoIP communication are mobile systems connected to different wireless network infrastructures (and both equipped with two or more wireless network interfaces). However, in this paper, in order to simplify our discussion, we consider the case in which a mobile host communicates with a fixed host as depicted in Fig. 2.

This paper is organized as follows. Section II summarizes related researches concerning mobility management and VoIP applications by comparing and contrasting them with the solution presented in the paper. Section III discusses the QoS requirements of VoIP applications and the possible conditions that may lead to their violations in a mobile WLAN context. Section IV describes the QoS mechanism we have developed. Section V presents some implementation details of that mechanism. In Section VI we report and discuss the principal results we have obtained from an experimental evaluation of our mechanism in a real wireless scenario. Finally, Section VII introduces some concluding remarks.

II. Related Work

In recent years, a body of research has investigated the challenges posed by the deployment of VoIP services in the context of WLAN networks.

This research activity has resulted in a large number of works. The authors in [9] describe a token-based scheduling scheme for fully connected WLANs that provides guaranteed priority access for voice traffic and, at the same time, a more accurate service differentiation for data traffic in WLAN environments. The scheme proposed in [9] does not use a central controller or a fixed logical ring in order to pass the token between mobile stations; rather, its token-based scheduling approach for wireless medium access is distributed and allows a mobile station to pass the token (there exist two tokens in the authors' system: one for voice traffic and one for data traffic) to other stations stochastically so as to achieve a proportional differentiation.

A resource allocation scheme for voice/data traffic over WLANs is described in [10]. The scheme consists of three principal mechanisms; namely, voice traffic multiplexing, which is achieved by combing mechanisms for controlled and contention-based accesses, deterministic access priority of voice that is realized with a minor modification to the IEEE 802.11e [11] concerning the contention behaviors of voice stations, and overhead reduction that is achieved by reducing the PHY and MAC layer overheads. This reduction is carried out aggregating the buffered voice packets from or to a voice station together and transmitting them by one MAC frame [10].

In [12] the authors propose a voice multiplex-multicast (M-M) scheme in order to support VoIP traffic in the WLAN context that coexists with further TCP traffic generated by other applications in execution in the WLAN. With that scheme, an AP multiplexes packets from different VoIP streams into one multicast packet for the transmission to the destination in order to reduce the overhead effect of VoIP over WLAN. In addition, the authors in [12] evaluate the interference problem between VoIP and TCP traffic at the buffer access of the AP and propose a priority-queuing solution in order to alleviate such interference.

A dynamically adaptable polling scheme is introduced in [13] in order to support voice traffic over WLANs. In particular, the authors in [13] propose a cyclic shift and a station removal polling scheme that is implemented on APs and does not require modifications to the mobile station medium access mechanism.

Finally, the IEEE 802.11e specification [11], that has been recently standardized, dictates how the current 802.11 MAC protocol has to be expanded to support applications with real-time QoS requirements (e.g., VoIP applications). Specifically, in order to provide QoS, the specification introduces priority schemes with the introduction of traffic categories.

These research works share the same objectives as our mechanism: all of them focus on investigating mechanisms that can support effectively the execution of VoIP applications in a WLAN environment. Nevertheless, the approaches proposed in these papers differ significantly from our approach. Our mechanism is implemented at the application layer and is based on a run time monitoring applied at the mobile station side only, in order to retrieve wireless link status information from the different ISO/OSI stack abstraction layers. This monitoring activity allows our mechanism to detect whether the wireless link used for the VoIP communication violates the interactivity and low packet loss QoS requirements. The previous cited works propose mechanisms to be deployed at the MAC layer (no cross-layer approach is taken into account). These mechanisms are not based on monitoring the QoS exhibited by the wireless link that connects a mobile station to the AP; rather, they principally focus on admission control and multiplexing techniques. In certain cases, they require modifications of the software running on the APs or the use of the IEEE 802.11e standard [11] that, to the best of our knowledge, is not widely used in existing WLAN networks, as yet. However, as our mechanism is applied at a higher abstraction layer than that in which the above works operate, it can be used in conjunction with such MAC layer mechanisms for scopes of reliability enhancements.

The research work that we believe comes very close to our solution is that described in [8] and, in more detail, in [14]. Those works describe a software architecture responsible for meeting QoS requirements of VoIP services in the WLAN context. Specifically, [8] introduces the Media Optimization Network Architecture (MONA) that uses multi-homing mechanisms in order to dynamically select an appropriate network medium for each application flow, according to the network conditions and the application flow QoS requirements. To this end, the authors propose a cross-layer approach that evaluates the QoS provided by the first hop from the host in the wireless access networks and its media selection. The approach is based on the implementation of an association layer that the authors have introduced in their architecture between the transport and the network layers in order to collect the status information of the wireless medium in one place. The association layer enables different techniques in order to maintain voice quality in WLANs. Among these techniques, they focus on the technique that enables a handover management [8], [14] with the aim to guarantee QoS in multiple WLAN environments. Handovers lead to packet losses with a resulting deterioration of the VoIP communication quality. In particular, the authors describe a mechanism to monitor the QoS of the link based on the number of frame retransmissions: with such a metric a mobile device can determine when the handover process should be started before packet loss actually occurs. The number of frame retransmissions is computed by taking into account the IEEE 802.11 specifications [5]: a data frame is successfully transmitted when the receiver sends back an ACK for that frame to the sender. When the data or ACK frames are lost, the sender retransmits the same data frames until the receivers sends back the ACK or until the number of retransmissions achieves a predefined threshold (this threshold ranges from four to seven depending on the frame size).

The research in [8] and [14] shares the same objectives and a number of similarities as our solution. To begin with, in both cases the deployed mechanisms are implemented between the transport and the network layers; they follow a cross-layer approach; that is, transport, network, and MAC layers interact one another in order to cooperatively use wireless link status information that are useful to detect possible QoS degradations. In addition, multi-homing capabilities are exploited both in [8] and in our QoS mechanism; this guarantees the use of the available wireless communication links of a mobile host when the condition of the link used for the communication impedes to meet the VoWLAN application QoS requirements.

Nevertheless, the two proposed schemes differ in both monitoring approach and objective. In particular, the authors in [8], at a predefined time interval, count the number of retransmissions on a wireless link in order to decide whether to perform a handover; in contrast, our approach is based on monitoring the transmission of each given UDP datagram, so as to detect whether the ACK frame is received within a predefined time threshold (specified by the application and according to its QoS requirements) and to retransmit that given packet; thus, our approach is driven from the application layer and allows a fine grained mobility and QoS management based on the management of each given datagram. This is the fundamental difference with the other approaches.

III. QoS requirements for VoIP applications

A VoIP application allows users at different end systems to carry out a voice conversation over the IP protocol. The mechanism is bidirectional and it works as follows. A VoIP application digitizes the user's voice, constructs UDP datagrams that contain approximately 20-40 ms of that digitized voice each, and transmits these datagrams through the network from a sender to a destination end system; the destination end system buffers the incoming datagrams until they are reconverted in voice for the end system user. The buffering process is necessary in order to compensate for variable network delays (i.e., jitter) and order the received packets.

Two principal metrics characterize the QoS requirements of VoIP applications; namely, sufficient interactivity and low packet loss.

The interactivity requirement is defined as the time delay elapsed between the sampling of the user voice at one end of the conversation and the playing out of that voice sample at the receiving end. In particular, the ITU-T G.1010 guidelines [4] recommend a one-way delay of up to 150 ms in order to use VoIP applications. Therefore, each UDP VoIP packet must be received at destination within the 150 ms time threshold in order to allow the end user to listen to the voice communication (note that this imposed limit can be hardly achievable in case of network congestion or long geographical distance between the end systems involved in the communication).

A voice packet is considered lost when it does not reach the destination end system within the interactivity threshold mentioned above. Hence, the percentage of packet loss should be maintained below 3% of the transmitted packets in order not to compromise the understandability of the entire voice conversation [4]. This limit considers the following observation for the packet loss distribution: in order to understand the entire sense of a voice conversation, a uniform distributed packet loss is less severe than the loss of more consecutive packets.

The QoS requirement analysis above indicates that it is crucial to carry out a monitoring activity that detects where and when packet losses occur. This activity allows one to control the percentage of packet losses: once the monitoring suspects a packet loss, a corrective action can be performed; i.e, the (suspected) lost packet can be retransmitted within a time threshold that is sufficient to meet the interactivity requirement mentioned above.

The next subsection discusses when violations of such QoS requirements can occur.

A. QoS requirement violations

In VoIP applications, QoS requirement violations can be due to different factors such as traffic bursts and router crashes. Traffic bursts are typically short duration events that lead to network congestion and delayed communications, in the presence of which routers can discard the packets they receive. In contrast, router crashes are typically long duration events that prevent packets to reach a destination end system until the routing tables, at the routers, are updated.

In case of mobile VoIP over Wireless LAN (VoWLAN) applications, the use of short-range wireless network technologies (the IEEE802.11 family) makes VoIP transmissions further troublesome, in particular in case of user movements. In a wireless environment obstacles can temporarily prevent the communication between mobile hosts and APs. Specifically, communication errors can be introduced as the datalink layer silently discards packets without any notification to the application layer.

In the presence of user movements, wireless communication links may become fully unavailable as, during the movement, the mobile host may lose the AP carrier. In this case, a handoff procedure is carried out in order to recover the communication: the mobile host has to select another AP, among those reachable, associate with it, and reconfigure itself in order to use the new selected AP (note that the handoff procedure can last a number of seconds that can be crucial in order to successfully meet the QoS requirements of VoWLAN applications).

In essence, we believe that errors in the wireless communication links due to obstacles or handoff procedures cannot be coped only with specific QoS techniques such as resource reservation or priority management, applied at each single wireless link, as wireless communication links can be temporarily or permanently unavailable. Rather, these errors can be the main cause of loss of consecutive packets that degrades a possible voice conversation, which is carried on those links. The errors are managed in different ways; the next subsection describes how to deal with them.

B. Notifications to the applications of QoS requirement violations

We distinguish between three principal types of error notifications, depending on the communication errors that can generate them. These notifications are summarized as follows.

1) **Notifications generated by the Network layer**: a number of communication errors are notified to the sender end system through ICMP messages. For instance, a router can detect that the destination end system is unreachable and notify the sender end system of this event via an ICMP message of type 0 (network unreachable error) or 1 (host unreachable error).

Specifically, an application that uses a TCP or UDP socket to send data can properly configure that socket (setting up the option IP_RECVERR through the system call *setsockopt*): if the datagram sending via the socket causes a communication error, the ICMP message, generated for that error and received by the sender end system, is queued in the socket buffer. In this way, the application can receive the error notification by simply reading the incoming message queue for the socket (this is technically carried out with the primitive *recvmsg,* using, as fourth parameter, tthe flag MSG_ERRQUEUE).

2) **Notifications generated by the destination end system application layer**: the router in the path between two end systems could not notify some types of communication errors. For instance, if a router discards an IP datagram because of network congestion, no notification from the router itself is sent to the sender end system. In this case, the applications can identify packet losses only by means of end-to-end algorithms that are based on both labeling the packets at the application layer and exchanging negative acknowledgments when specific timeouts expire.

This type of notifications, in which the sender end system is actually aware of a communication error only when the entire path from the destination to the sender is covered again, can suffer from high delays (in particular when the network latency is high) and prevent the application to perform a timely error recovery. For these motivations, the periodical reports that are performed by control protocols such as RTCP [15] are not effectively usable in order to recover from communication errors.

3) **Failed notifications from the wireless MAC layer**: in IEEE 802.11/b/g/n [5, 6, 7], the receiver sends back an acknowledgment to the sender in order to confirm that a unicast frame is successfully transmitted to an AP. When data frames or ACK frames are lost, the sender MAC layer retransmits the frame over the wireless link until the number of retransmissions reaches a predetermined retry limit that is a maximum of seven retries. After that limit, the MAC layer silently discards the frame, without generating any notification. The discarded frame never reaches the destination end system: its loss can be identified by the application layer protocols only when it could be too late to perform recovery procedures, as described in 2) above.

In the context of this work, we focus on the third type of communication errors. In particular, in this paper we describe a QoS mechanism that timely propagates the notification that a packet has been locally discarded up to the VoIP transmitting application, in order for that application to properly and timely execute an error recovery procedure.

IV. Design Issues

The QoS mechanism we have developed is named Robust Wireless Multi-Path Channel (RWMPC).and is depicted in Fig. 3. It guarantees VoWLAN applications that, under some constraints described in the following sub-section *A* (*Interactivity Model*), UDP datagram losses will be at most 3% of the transmitted datagrams, and communication delays will be below 150 ms, thus enabling an interference-free interactivity between communicating parties using VoIP. The RWMPC mechanism is based on the ABPS abstraction model, described in the introduction, that dictates to use simultaneously all the NICs available in a MN and to select for each datagram the most suitable NIC.

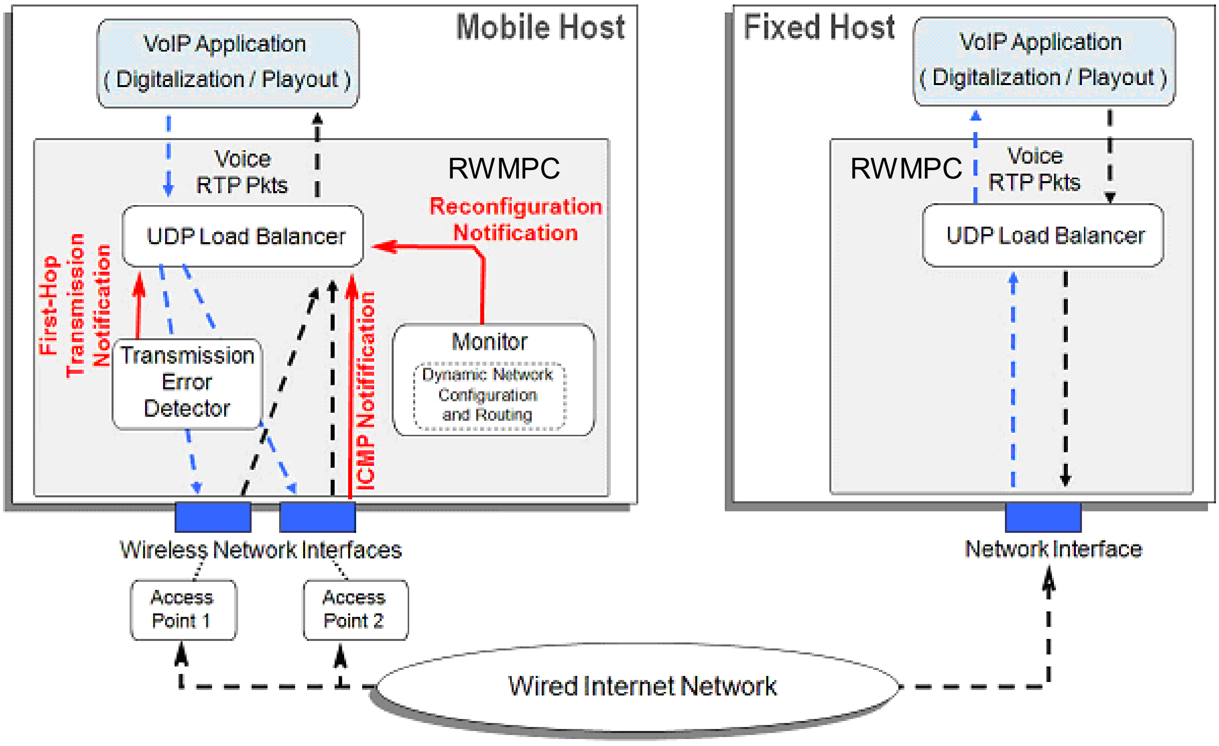

Figure 3. The overall RWMPC middleware system.

The RWMPC mechanism uses a cross-layer approach in order to timely propagate, from the MAC layer up to the application layer, the information concerning the transmission status of UDP datagrams generated by a VoIP application. This cross-layer approach allows the application layer to carry out a timely error recovery procedure, and decide which wireless interface, among those available on the mobile host, is to be used for the transmission of the successive UDP datagrams. Note that an end system receiving a UDP datagram is to be made aware of which wireless interface has been used by the source mobil) host of that datagram so as to respond to it.

Essentially, the RWMPC is a middleware deployed in both the end systems of a VoIP communication. The mobile and fixed hosts run the VoIP client application that exhibits an interface for use by the end-user, digitizes the end-user voice, and passes the voice packets to the RWMPC middleware system. The middleware at the mobile sender side is responsible for encapsulating voice packets in UDP datagrams and transmitting (and, if necessary, retransmitting) those datagrams to the destination end system. The middleware at the fixed destination end system receives the datagrams, orders them, and passes the datagrams to the application for the final playout.

The middleware deployed in the mobile host executes on top of a Linux operating system and consists of the following components (see Fig. 3):

1) **Transmission Error Detector** (**TED**): this component is the most important one of the RWMPC middleware, as it monitors whether or not UDP datagrams have been successfully received by the APs. It also notifies the UDP Load Balancer component (see below) of UDP datagram rejections;

2) **UDP Load Balancer** (**ULB**): this component receives error notifications from the TED component and decides if lost UDP datagrams are to be retransmitted. In addition, based on the notifications received by the TED component, it decides which wireless interface, among those available on the mobile host, is to be used for sending the successive UDP datagrams;

3) **Monitor**: this component is responsible for monitoring and configuring the wireless interfaces homed in the mobile host, and reporting to the ULB component which interfaces are actually active and configured. In more detail, the **Monitor** component operates as a separate application that configures dynamically the wireless network interfaces and the routing rules. It communicates with both the Linux kernel and UDP Load Balancer component. Specifically, it notifies this latter component with a Reconfiguration Notification (see Fig. 3) when a wireless network interface has been either fully configured (and can be used), or disabled as a consequence of a communication error that has been detected.

The **Transmission Error Detector** (**TED**) component operates at the MAC layer of the IEEE 802.11b/g/n/e [5, 6, 7, 11] protocol; its principal responsibility is to monitor each single UDP datagram sent to the fixed destination end system over an active and configured wireless link in order to detect whether that datagram has been successfully received by the AP or discarded by the MAC layer. TED notifies the UDP Load Balancer component of the status of this transmission via the "First-hop Transmission Notification" of Fig. 3. Specifically, TED notifies the UDP Load Balancer component by queuing an error message in the message queue of the socket that has been used to send the UDP datagram.

The **UDP Load Balancer** (**ULB**) component receives the RTP voice packets generated by the VoIP application, encapsulates them in UDP datagrams, and sends the datagrams to the destination end system. To this end, ULB uses a UDP socket for each active wireless network interface of the mobile host; each interface has been properly configured by the Monitor component. When it is created, each UDP socket is connected to a wireless interface using the *bind* primitive; this permits to utilize only that interface.

ULB receives three types of notifications (the non dashed lines in Fig. 3): a first type of notification comes from the Monitor component. This component can inform ULB that either a new network interface is available or a given active interface has been disabled. For each new available interface, ULB generates a UDP socket and binds that socket with the new interface in order to send and receive UDP datagrams through the interface. In contrast, for each interface that has been disabled, ULB closes the corresponding UDP socket and considers lost the UDP datagrams that have been sent through that socket and for which no "First-hop Transmission Notification" from the TED component has been received.

ULB receives other two types of error notifications with the *recvmsg* system call, using the MSG_ERRQUEUE parameter. Specifically, one type of notification is sent by the ICMP protocol to a UDP socket in order to notify the ULB component that some UDP datagrams have been lost along the end-to-end path. The other type of notification is generated by the TED component in order to inform ULB that a given UDP datagram has been either successfully received by the AP or permanently discarded.

Based on these three types of notifications, ULB decides if a lost UDP datagram has to be either retransmitted using another wireless network interface among those available on the mobile host, or permanently discarded. In addition, based on both the monitoring notifications of the sent UDP datagrams and the ICMP notifications, the ULB component selects the wireless network interface to be used for sending successive UDP datagrams. Subsection B describes the algorithm used for selecting a wireless interface.

The RWMPC middleware system deployed in the fixed host (see Fig. 3) is simpler than that in the mobile host, described so far; it consists of a simplified version of the ULB component above that controls only one wired network interface. The ULB component at the fixed host assumes that the transmitted datagrams are not discarded in the first wired hop; hence, it does not need to be notified in case or not the wired Internet network has successfully received the UDP datagram. Finally, its principal activity consists of recording the IP address of the mobile host wireless interface from which it has received the last UDP datagram. That IP address is used for sending successive response UDP datagrams to the mobile host.

A. Interactivity model

We assume that the network environment consists of a wireless link and a wired path. Hence, let D' be the probability that an IP datagram is lost in the wireless communication link, D'' be the probability that an IP datagram is lost in the wired path, and *f* the probability density function of the datagram arrival time; that is, *f(t)* is the probability that a voice packet has been received at the destination end system exactly at the time instant t.

Let T=150 ms be the maximum time within which the packet voice has to be delivered to the destination end system.

We assume that the mobile host is equipped with two wireless interfaces that are associated with two different APs; these two APs are connected to the fixed destination end system via a wired network. Let T_R be the timeout that the ULB component uses to trigger the voice packet retransmission once expired (no successful transmission notification is received in this case by the ULB component).

If the mobile host is equipped with one wireless adapter only, and does not deploy our RWMPC middleware, the probability that a voice packet is lost, or has been received at destination after the predetermined T time threshold (late arrival), is the following:

$$D'+(1-D')\left(D''+(1-D'')\int_{T}^{+\infty} f(t)dt\right) \tag{1}$$

In contrast, suppose that the mobile host does deploy our RWMPC middleware system, if the voice packet is lost along the wired path or arrives at destination after the predetermined T time threshold, the TED component cannot detect the loss and consequently retransmit that packet: the packet is thus permanently lost. In this case, the probability that the packet is lost is given by equation (2) below.

$$(1-D')\left(D''+(1-D'')\int_{T}^{+\infty} f(t)dt\right) \tag{2}$$

If the packet is lost in the wireless link (with D' probability), TED detects the loss and retransmits the packet. If the TED component retransmits the packet after T_R the packet takes T-T_R time before arriving at the destination end system in time. However, the packet can be still lost along the path. In this case, as the packet cannot be retransmitted, the packet is permanently lost. Hence, the probability to lose permanently the packet during the retransmission is given by the following formula:

$$D'+(1-D')\left(D''+(1-D'')\int_{T-T_R}^{+\infty} f(t)dt\right) \tag{3}$$

In essence, the probability that a packet is lost or arrives after the predetermined T time threshold is:

$$(1-D')\left(D''+(1-D'')\int_{T}^{+\infty} f(t)dt\right)+$$

$$+D'\left(D'+(1-D')\left(D''+(1-D'')\int_{T-T_R}^{+\infty} f(t)dt\right)\right) \tag{4}$$

The difference between (1) and (4) is the percentage of packets that are not lost, using our RWMPC middleware. This difference is given by the following formula:

$$(1)\text{-}(4) =$$

$$D'\left(1-\left(D'+(1-D')\left(D''+(1-D'')\int_{T-T_R}^{+\infty} f(t)dt\right)\right)\right)$$

$$= D'P_{pktArrived} \tag{5}$$

where

$$P_{pktArrived} = \left(1-\left(D'+(1-D')\left(D''+(1-D'')\int_{T-T_R}^{+\infty} f(t)dt\right)\right)\right)$$

$P_{pktArrived}$ in (5) represents the percentage of packets that arrive at destination within (T-T_R) during the retransmission phase. Hence, the percentage of packets that are not lost augments when both the percentage of losses on the wireless link and the time for the packet to reach the destination end system increase.

In essence, if the percentage of losses on the wireless link is high, our RWMPC middleware is significantly effective.

This result is confirmed by the following numerical example. We assume that the probability to lose packets in a wireless link is D'=10%, the percentage of losses on the wired network is D''=0.5%, the maximum time limit for a packet to arrive at the destination end system is T=150 ms, and the timeout for packet retransmission is T_R=30 ms. In addition, we assume that the probability density function of the transmission time is a lognormal function with mean equal to 0.078 and standard deviation equal to 0.28 (with this function, approximately 1% of packets arrive at destination after 150 ms). Hence, the probability to lose a packet or to receive it after the specified T time threshold (i.e., late arrival) as computed in 1) is approximately 11.3%, which exceeds the ITU-T requirement (3%). In contrast, using our RWMPC middleware system, this probability, as computed in 4), is approximately 2.9%; that is, the ITU-T requirement is met.

B. Wireless network interface selection at the mobile host

When using certain wireless network interfaces, the TED component does not produce notifications in case datagrams have been discarded by the network interface itself. The motivation for this behaviour can be explained as follows. Some wireless network interface firmware, available on the market, does not actually report to the MAC layer that a given frame has been discarded; rather, most of them inform the MAC layer when a frame has been successfully delivered to the AP. Therefore, the ULB component assumes that a datagram has been discarded by the wireless link when either a lost datagram notification is received from TED, or a timeout (that we have fixed to 30 ms) expires after the datagram has been delivered to the socket without any notification from TED of successful datagram transmission.

ULB maintains a list of the datagrams that are to be transmitted, in the order they have been received from the application layer. The same order is maintained for their transmission through the most suitable wireless network interface, using the UDP socket that has been bound to that interface. The algorithm that selects the most suitable wireless network interface is very simple. From the standpoint of the ULB component, each UDP socket can be in three states, only: WORKING, SUSPECTED, and DISABLED. The initial state is WORKING. When notifications of the status of a transmission are raised, the UDP socket moves from state to state. If a UDP socket is in a WORKING state and it receives either an ICMP error or a Reconfiguration Notification that notifies the socket of the network interface unavailability, the UDP socket makes a transition to the DISABLED state. If a UDP socket is in a WORKING state and either the 30 ms timeout mentioned above expires or the socket receives the First-hop Transmission Notification that a UDP datagram has been discarded, the UDP socket makes a transition to the SUSPECTED state. In contrast, if the UDP socket is in a SUSPECTED state and it receives either UDP datagrams from the fixed end system or a First-hop Transmission Notification that the UDP datagram has been successfully transmitted to the AP before the 30 ms timeout expires, the UDP socket makes a transition to the WORKING state.

Based on these states, the ULB component selects the UDP socket to be used for the transmission as follows.

When ULB needs to transmit a given UDP datagram, it selects a socket that is in the WORKING state. If no WORKING socket is available, ULB selects a SUSPECTED socket. If no SUSPECTED socket is available the datagram is discarded.

After a given UDP datagram has been sent, the ULB component sets the earlier 30ms timeout for that datagram. If no notification of successful datagram transmission is received from TED before the timeout expires, the datagram is retransmitted using a different network interface (and its associated socket). The socket selection process is repeated excluding the socket that has been previously used. If no socket is available the datagram is discarded.

V. IMPLEMENTATION

A prototype of the described RWMPC system has been implemented using as mobile host a laptop equipped with the Linux operating system (Gentoo distribution) and modifying the kernel version 2.6.27.4. In fact, the implementation of the TED component includes a novel system call, named sendmsg_getID, that delivers, for successive transmissions, a UDP datagram to the network layer and returns to the application layer the unique identifier of the IP datagram that encapsulates that UDP datagram; moreover, we have modified the kernel's mac80211 module so as to detect the time instant in which the wireless adapter driver reports whether a given frame has been transmitted to the AP or discarded.

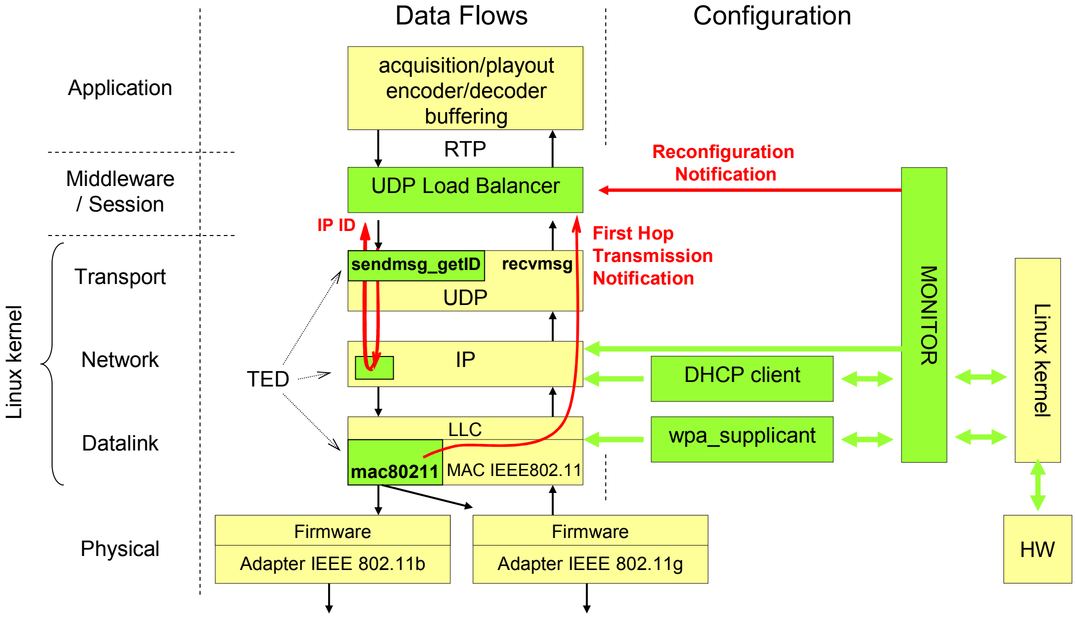

Figure 4. RWMPC's components at the MN.

The ULB component has been implemented at the middleware layer and the Monitor as a set of separate applications. Fig. 4 shows the overall RWMPC architecture deployed at the mobile host side. The darker rectangles in this figure represent the RWMPC software components that we have designed and implemented; the next subsections describe them in more details.

A. The Monitor

RWMPC uses already existing network infrastructures, which can be administered by different organizations. We assume that each network domain provides its mobile clients with a DHCP service that distributes IP addresses to those clients. A separate application, named **Monitor**, configures dynamically the wireless NICs and the routing rules, and notifies the ULB component of which wireless NIC has been completely configured and can be used (the message labeled "Reconfiguration Notification" in Fig. 4). In particular, the Monitor communicates with the Linux kernel by means of a datagram-oriented Netlink socket: when a network adapter changes its status, the kernel informs the Monitor, which in turn activates the adapter reconfiguration procedures.

The operations that allow each wireless NIC in the mobile host to detect the presence of an AP, and configure its own data link layer to communicate with that AP are implemented at the datalink layer. In addition, the data link layer is responsible for managing the NIC that loses the AP carrier: that adapter cannot communicate and thus must be disabled. All these operations have been implemented as a separate process, reusing a widely used, free, and open-source application called **wpa_supplicant**. Wpa_supplicant encloses all those functionalities related to the authentication and security of wireless networks. As to the authentication functionality, it requires the use of a configuration file, which contains the information necessary to enable wpa_supplicant authentication in each visited networks. As to the security functionality, wpa_supplicant hides the complexity of different security mechanisms. Simply stated, in the IEEE 802.11 scenario, wpa_supplicant drives the wireless adapter in scanning the radio channels to detect the APs of the wireless networks of which it knows the authentication information. If more APs are available, wpa_supplicant selects the one that provides the best radio signal, and executes the authentication procedure with that AP. We have implemented a minor modification of the wpa_supplicant AP selection process in order to avoid that more wireless NICs of a given mobile host select the same AP and thus differentiate the available paths towards the destination end system.

After wpa_supplicant has configured the data link layer of a wireless NIC, by associating it to a given AP, the kernel informs the Monitor that starts a DHCP client. The DHCP client requests an IP address, a netmask, and an IP gateway to the DHCP server of the visited network. When the DHCP client terminates successfully the configuration of the network layer of a NIC, the Monitor sets up a new rule and a new routing table (invoking the "ip rule" and "ip route" commands) in order to enable the dynamic routing via this NIC. These rule and routing table impose that IP datagram having the same source IP address as this NIC are routed through it, regardless of their IP destination address. Once this NIC has been configured, it can be used by the ULB component.

Finally, when the NIC loses the AP carrier, the adapter's driver interrupts the kernel that contacts the Monitor in order to disable that NIC and delete the routing rule of this NIC; the NIC will be unavailable until a new reconfiguration of it is performed.

B. The Transmission Error Detector

The TED component consists of two parts; namely, a first part implemented at the transport and network layer, and a second part implemented at the MAC sub layer of the datalink layer, as depicted in Fig. 4. When ULB sends a UDP datagram to the destination end system, it needs to be informed whether that datagram has been either received by the AP or discarded. In order to achieve this objective, TED implements two tools:

1) a novel system call, named *sendmsg_getID*, included in the UDP and IP Linux kernel modules. This system call extends the behavior of the existing sendmsg system call, which is responsible for starting the transmission of a UDP datagram to a given destination using a given UDP socket. In particular, a UDP datagram is delivered to a local system buffer of that UDP socket; the routing module decides the NIC to be used for the transmission, and afterwards the sendmsg function returns the control to the caller. In addition, our *sendmsg_getID* extension returns an integer value to the caller; the value is the ***id*** field of the IP header that precedes the UDP datagram. The *id* is a (temporary) unique IP datagram identifier. The ULB component uses the sendmsg_getID function and maintains the unique *id* of each UDP datagram sent to the destination end system, waiting for a notification of successful or failed transmission from the MAC layer part of the TED component. This notification contains the *id* of the UDP datagram;

2) a notification system, which has been implemented in the MAC layer module of the Linux kernel wireless stack. After a socket sends a UDP datagram, that datagram is encapsulated inside an IP datagram and a datalink frame, and then delivered to the firmware of the wireless NIC. The firmware carries out asynchronously the transmission to the AP and eventually returns the frame transmission outcome to the MAC layer. The TED component receives that outcome from the firmware: if the frame contains a UDP datagram, TED extracts the outcome, finds the socket that has been used to send the datagram, and informs the sending socket by delivering a particular message in the socket message queue. Typically, each socket implements an internal queue of messages in which the ICMP error messages sent to that socket are buffered. The application may enable the use of that error queue by configuring the socket (this is technically carried out by invoking the setsockopt system call with the IP_RECVERR flag). In our implementation, we have extended the use of the error queue by introducing a new type of message (i.e., the IP_NOTIFY

message). This message contains a) the *id* of the IP datagram, b) the outcome of the IP datagram fragment transmission, c) the length of the fragment, d) the more fragment field and, e) the offset field of the IP datagram fragment.

ULB uses the earlier notification system, enables the use of the socket message queue, and waits for IP_NOTIFY messages. These messages are read using the recvmsg system call with the MSG_ERRQUEUE flag in order to understand whether a given UDP datagram has been sent or discarded. Unfortunately, the firmware of some wireless network adapter (for instance, the WUSB54G Linksys adapter that we have used in our implementation) notifies successful transmissions but does not notify failed frame transmissions. Thus, the ULB component must implement a timeout-based approach for each UDP datagram in order to decide whether that datagram is to be retransmitted.

It is important to point out that we have implemented the TED component on Linux kernel version 2.6.27.4. In fact, starting from the 2.6.24 version, the Linux kernel implements a complete IEEE 802.11 protocols stack and, in particular, it provides the implementation of the MAC layer in the mac80211 module (see Fig. 4 above). Thus, each wireless NIC driver relies on that module and does not need to implement its own wireless stack (this approach allows us to use all the recent wireless network interfaces available on the market).

C. The UDP Load Balancer

This component is responsible for the transmission of the voice packets via all available wireless network adapters, and for the recovery from possible packet losses. It receives error notifications from the TED component and decides if lost UDP datagrams are to be retransmitted, following the algorithm for UDP socket selection described in subsection IV.B. In addition, based on the notifications received by TED, it decides which wireless interface, among those available on the mobile host, is to be used for sending the successive UDP datagrams. ULB receives information about all the available network adapters from the Monitor and creates a UDP socket for each of them; to this end, it uses the bind() system call in order to assign the IP address of the chosen adapter to the associated socket. Thus, all IP datagrams belonging to that socket have their IP adapter address as their source IP address. Owing to the routing rules created by the Monitor, all IP datagrams belonging to a given socket are routed through the adapter to which that socket has been bound. In other words, the ULB component may select the wireless adapter to be used to send a datagram by selecting for the transmission the associated UDP socket. ULB has been implemented so as to avoid busy waiting: the implementation is mainly based on a loop driven mechanism that makes use of the system call select. The select system call waits, without consuming CPU time, for the sockets to change their status, and then performs according to the new socket state. In particular, it processes notifications belonging to the TED and Monitor components, sends, and receives RTP voice packets.

VI. Experimental Evaluation

We have carried out an experimental evaluation of a RWMPC prototype. The objectives and the principal results of this evaluation are summarized as follows. The main objective was to show the RWMPC ability to effectively react to the changes in the availability of the communication resources, and deliver the VoIP frames to the destination within 150 ms, with an acceptable level of packet losses, so as to provide a delivery service suitable for VoIP applications.

We have conducted the experimental evaluation in the urban area of our Department of Computer Science at the University of Bologna. A laptop running the Mobile Side (MS) of the RWMPC covered a route of 80 meters within this area. The RWMPC is equipped with two IEEE 802.11g [6] wireless interfaces that can be associated with two APs available in the urban area.

The experimental scenario we have used is depicted in Fig. 5. As shown in that figure, along this route there exist five IEEE 802.11g APs: two APs (i.e., AP1 and AP2) exhibit a transmission signal that is strongly and abruptly obscured by walls and reinforced concrete columns; however, their transmission signals are captured by the MS so that the MS wireless interfaces can be associated with them.

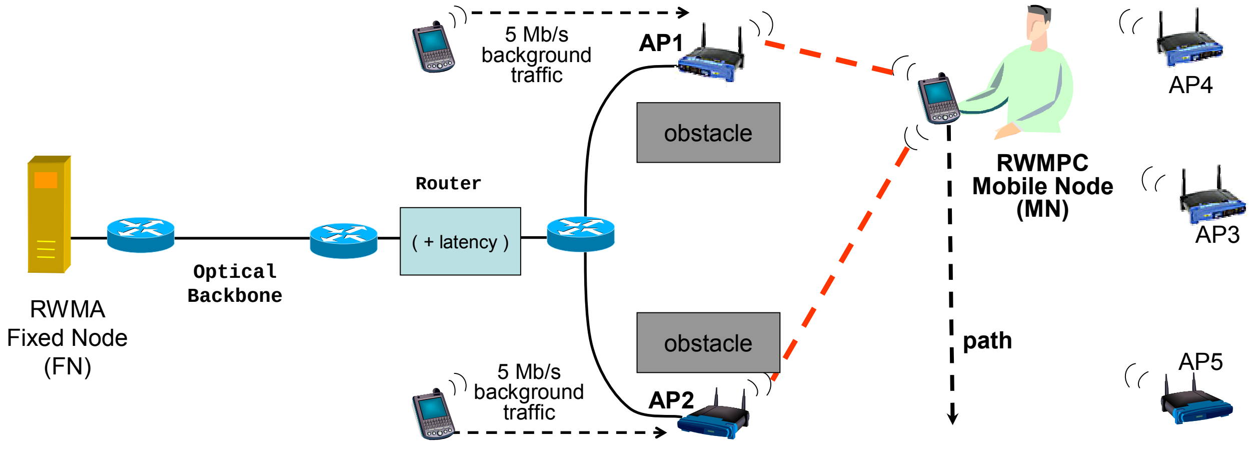

Figure 5. Experimental Scenario

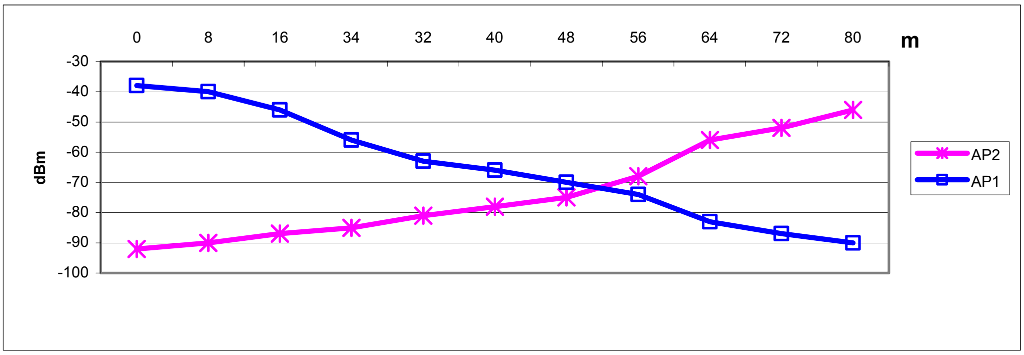

Figure 6. Access Point signal strength.

The remaining three APs (i.e., AP3, AP4, and AP5) are not used by the MS and operate on channels that are adjacent to those on which the APs associated with the MS operate, thus producing inter-channel interferences.

The access point AP1 is located near the beginning of the MS path; it operates on the channel 1 and its signal is partially covered by the above mentioned obstacles. Hence, at the beginning of the path it provides the MS with a power of -37dBm; the power progressively decreases to -65dBm in the middle of the path and to -90dBm near the end of the path. In contrast, the access point AP2 works on channel 6 and is located close to the end of the path the mobile host covers. It is not accessible at the beginning of the path and it provides the MS with a power of approximately -92dBm; this power becomes -77dBm and then approximately -45dBm at the end of the MS path (see Fig. 6). In other words, the access point AP2 is unavailable at the beginning of the MS path whereas the access point AP1 is unavailable at the end of the path.

Both the APs are affected by additional background traffic of approximately 5Mbps. This traffic is generated by two mobile hosts depicted in Fig. 5. The Fixed Side (FS) of RWMPC is located at the decentralized University of Bologna's departments in Cesena, approximately 80 km away from Bologna. A router is located at the beginning of the backbone. The router is a FreeBSD machine that uses the *ipfw* services in order to intercept datagram that are sent from and received by the MS. We have configured the *ipfw* service at the router so that it buffers the datagram for a variable time, before transmitting the fragments to the destination. That buffering delay introduces, in the communications through the backbone, a latency confined in the range of [70;80] ms in the tests we have carried out (see below). This choice allows us to analyze the RWMPC system behaviour in a realistic scenario.

A probing program has been written to simulate the traffic of a VoIP session, similarly to the approach adopted in [16]. The program sends a continuous sequence of frames through the two ULB components from the FS to the MS. The payload size of each frame is 160B and a frame is sent every 40 ms. This scenario emulates a VoIP session of 32 Kbps with no silence suppression. We recorded the delivery time of each frame from the MS to the FS and vice versa.

In the tests we have carried out, the communication errors were principally concentrated in the area in which the AP1 became unavailable (due to its poor signal strength). In this case, the RWMPC selects for the transmission the wireless interface that makes use of the other available access point AP2. In the time period in which the AP1 becomes unavailable, and before the RWMPC definitively selects the use of the AP2 only (that is approximately 10 seconds and named *transition period*) less than 10 VoIP packets are lost in the direction from FS to MN, as depicted in Fig. 7. All the graphs of our experimental evaluation show the results we have obtained in the transition period.

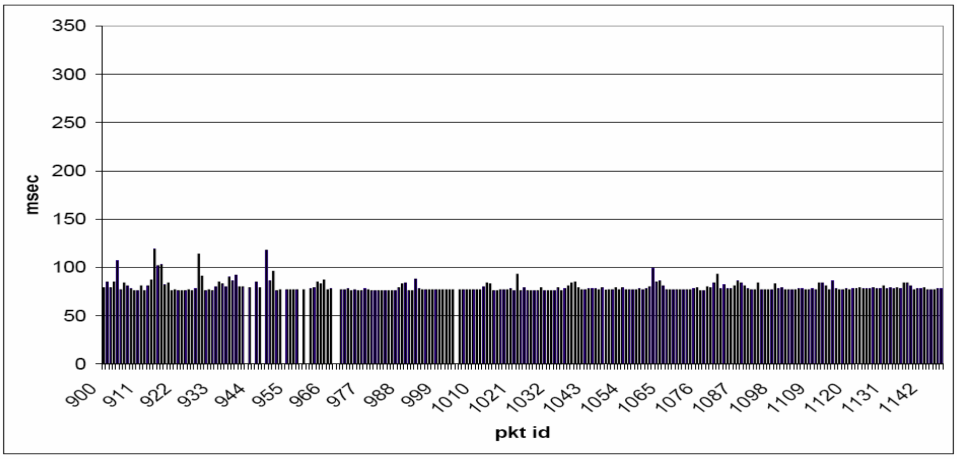

Figure 7. delay of the packet received at the mobile side

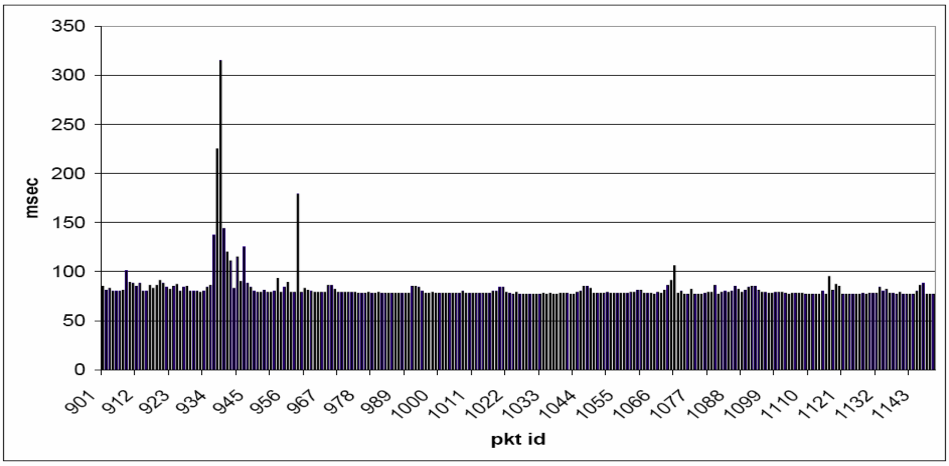

Figure 8. delay of packets received at the fixed side

Specifically, Fig. 8 depicts the delay of the packets received at the RWMPC fixed side. In this case the mobile host detects that six packets have not been successfully transmitted to the fixed side and retransmits those packets on the second wireless interface it owns associated with AP2. Afterwards, the mobile host permanently uses AP2 for its future transmissions. Owing to this RWMPC behavior, the fixed host receives all the packets: only three of them arrive after the 150 ms time threshold we have imposed and are considered lost (Fig. 8). Two of these three lost packets are consecutive and produce a voice loss equal to 80 ms.

At the RWMPC mobile side, during the transition period, only nine packets are lost in the wireless path, as shown in Fig. 7. Among these lost packets, two groups consist of two consecutive packets, only; the remaining lost packets are isolated. In contrast, the packets that reach the destination end system have been received in time, i.e., within the 150 ms time threshold that represents the interactivity QoS requirement.

VII. Concluding Remarks

In this paper we have introduced the Always Best Packet Switching (ABPS) model and described the RWMPC mechanism that we have designed and developed in order to meet interactivity and low packet loss QoS requirements of VoWLAN applications. The RWMPC, uses a cross layer approach in order to monitor the first hop in the voice communication that connects a mobile host to an AP, and to enable, if necessary, VoIP packet retransmissions on an alternative hop, available through one of the wireless interfaces homed in the mobile host. The monitoring activity is necessary in order to detect whether or not the hop in use violates the above mentioned QoS requirements and decide how the VoIP packet retransmission is to be performed.

From our interactivity model we can state that, provided that (i) mobile devices are equipped with two or more wireless network interfaces, (ii) the probability to lose packets on the wireless links does not exceed the 10% of the transmitted packets, and (iii) no more than 1% of the transmitted (and not lost) packets are received at destinations after 150 ms, the RWMPC can meet the interactivity requirement by guaranteeing that transmission delays are maintained below 150 ms, and the low packet loss requirement by ensuring that the loss is maintained below the 2.9% of the transmitted packets, whereas without our mechanism the loss is about 11.3%. We have carried out an experimental evaluation of our system in a real wireless scenario; this evaluation confirms both the adequacy of our model and the effectiveness of our developed RWMPC middleware.

References

[1] Boing wireless, http://www.boingo.com/, home page, 2008.

[2] Fonera, http://www.fon.com/, home page, 2008.

[3] E. Gustafsson and A. Jonsson, "Always Best Connected,, in *IEEE Comm. Mag.*, vol. 10, no. 1, Feb. 2003, pp. 49–55.

[4] ITU-T Recommendation G.1010, "End-user multimedia QoS categories," November 2001.

[5] IEEE Std. 802.11b-1999, "Higher-Speed Physical Layer (PHY) extension in the 2.4 GHz band," IEEE Standard for Information Technology, 1999.

[6] IEEE Std. 802.11g-2003, "Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band", IEEE Standard for Information Technology, 2003.

[7] IEEE Std. 802.11n-2007, "Higher Throughput Improvements using MIMO", IEEE Standard for Information Technology, 2007.

[8] H. Koga, S. Kashihara, Y. Fukuda, K. Iida, Y. Oie, "A quality-aware VoWLAN architecture and its quantitative evaluations", *IEEE Wireless Communications*, 13(1), February 2006.

[9] P. Wang, W. Zhuang, "A Token-Based Scheduling Scheme for WLANs Supporting Voice/Data Traffic and its Performance Analysis," in *IEEE Transactions Wireless Communications*, 7(4), April 2008.

[10] P. Wang, H. Jiang, W. Zhuang, "Capacity Improvement and Analysis for Voice/Data Traffic over WLANs," in *IEEE Transactions on Wireless Communications*, 6(4), April 2007.

[11] IEEE Standards Association, IEEE 802.11e, available at http://standards.ieee.org/getieee802/download/802.11e-2005.pdf, 2008.

[12] W. Wang, S.C. Liew, V. O. K. Li, "Solutions to Performance Problems in VoIP Over a 802.11 Wireless LAN," in *IEEE Transactions on Vehicular Technology*, 54(1), January 2005.

[13] E. Ziouva, T. Antonakopoulos, "A dynamically adaptable polling scheme for voice support in IEEE802.11 networks,"

in *Computer Communications* 26(2), pp. 129-142, February 2003.

[14] S. Kashihara, Y. Oie, "Handover management based on the number of data frame retransmissions for VoWLAN," in *Computer Communications (Elsevier)*, pp 3257-3269, 30 January 2007.

[15] Network Working Group, "RTP: A Transport Protocol for Real-Time Applications," July 2003.

[16] A. da Conceicao, L. Jin, D. A. Florencio, F. Kon, "Is IEEE 802.11 ready for VoIP?," Proc. of *8th Workshop on Multimedia Signal Processing*, October 2006.

Vittorio Ghini received the "Laurea" degree in Computer Science from the University of Bologna (Italy) in 1997 and the Ph.D. degrees (2002) in Computer Science from the University of Bologna.

Since 2005 he is a Research Associate at the Computer Science Department of the University of Bologna. His research interests include distributed multimedia systems, middleware protocols for QoS over IP networks, and dynamic multi-homing management.

Giorgia Lodi received the "Laurea" and Ph.D. degrees in Computer Science from the University of Bologna (Italy).

She was a research associate of Computer Science at the University of Newcastle upon Tyne (UK) in 2002 and at the University of Bologna (Italy) from 2002-2008. She is currently a research associate of Computer Science at the University of Rome "La Sapienza" where she works in the context of the EU funded Project "CoMiFin". Her research interests include distributed systems, middleware, application server technologies, clustering techniques, SLA management and security.

Fabio Panzieri received the Laurea degree in Computer Science from the University of Pisa (Italy) in 1978, and the PhD in Computer Science from the University of Newcastle upon Tyne (UK) in 1985.

He is a Professor of Computer Science at the University of Bologna (Italy). His research interests include middleware architectures, large scale reliable distributed systems, distributed real time systems, and communication protocols.