Wireless Pers Commun
DOI 10.1007/s11277-008-9630-y

Meeting Interactivity Requirements in Mobile E-Witness: An Experimental Study

Vittorio Ghini · Giorgia Lodi · Stefano Cacciaguerra · Fabio Panzieri

Abstract This paper describes the design and development of a session-layer load balancing mechanism and its deployment in a mobile surveillance system named Mobile E-Witness (MEW), which we have developed (Ghini et al., *Multimedia Tools and Applications, 37*(3), 293–318, 2008). The load balancing mechanism proposed in this paper allows MEW to effectively meet interactivity requirements and it is based on an "early retransmission" technique that exploits the overall bandwidth provided by a number of heterogeneous (broadband and metropolitan area) wireless adapters incorporated in MEW. This technique anticipates a suspected unavailability of the adapters in order to avoid the effects of network congestion and guarantee continuity of the communication service and support for multimedia services such as IP telephony. We have carried out an experimental evaluation of the MEW prototype in order to assess its effectiveness in meeting interactivity and responsiveness requirements. This paper summarizes our design and the principal results we have obtained from the evaluation mentioned above.

Keywords Resource and mobility management · Wireless networks · Sensor networks · Multi-homing · Session-layer load balancing · Multimedia data streams

1 Introduction

Nowadays, we are witnessing an increasing deployment in our cities of wireless communication technologies (e.g., IEEE 802.11, mobile WiMAX [12], cellular technologies) that are used to support a variety of services including multimedia services such as mobile audio and video streaming.

Typically a wireless scenario suffers from limitations like scarce communication resources in terms of bandwidth provided by the access points, and lack of continuity of the communications between access points and mobile devices. These limitations can be exacerbated

V. Ghini · G. Lodi (✉) · S. Cacciaguerra · F. Panzieri
Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy
e-mail: lodig@cs.unibo.it

in the specific metropolitan context in which multiple obstacles can reduce the transmission range of the access points, and a large number of concurrent users can consume the available bandwidth.

In order to overcome these limitations and effectively deliver multimedia services such as those earlier mentioned, a possible solution is to allow the inter working of heterogeneous wireless technologies, thus enabling the cooperation among different, and possible independently managed, wireless network infrastructures. Within a cooperative wireless scenario, an advanced resource and mobility management mechanism is to be adopted in order to meet multimedia application Quality of Service (QoS) requirements (e.g., responsiveness, reliability, availability, continuity of the communications).

In line with this scenario, we have recently assessed the requirements of a multimedia application termed *Mobile E-Witness* (MEW) that we have developed for scopes of surveillance and protection of the citizen safety in metropolitan areas [7].

MEW enables the acquisition and remote storage of multimedia (i.e., audio and video) data streams. In MEW, a mobile device termed Mobile Sensor (MS) can be worn by public officers such as policemen and health care workers in order to record the events these officers witness while on duty. MEW transmits these events to a remote data storage termed Control Center (CC) for future replay. A recorded event can be then used as an impartial testimony to resolve possible disputes concerning the relative responsibilities of the participants to that event (including the officers themselves).

In order to collect and transmit multimedia data the MSs exploit the infrastructure consisting of wireless hotspots, and wired communications behind them, that can be available in metropolitan areas. This infrastructure can be partly privately and partly publicly owned, independently managed, and provides a best effort communication service.

As demonstrated by our recent MEW experimental evaluation [7], carried out in Bologna (Italy) using IEEE 802.11 b/g technologies only, we have been able to achieve the following five principal results: MEW can (1) tolerate temporary disconnections with the wireless access points so as to guarantee highly available communications; (2) ensure enough bandwidth for multimedia data transmission; (3) ensure end-to-end reliability by implementing retransmission techniques that guarantee the delivery of all the audio/video data frames to destination; (4) limit, within a couple of seconds, the elapsed time between the acquisition of each audio/video frame and the storage of that frame to the remote CC; and finally (5) limit the power consumption of the MS transmission so as to reduce the electromagnetic radiations absorbed by the human operator.

However, the MEW architecture described in [7] exhibits a principal shortcoming that concerns its level of responsiveness, measured as the maximum frame storage delay. This metric represents the upper limit of the overall one-way delay that may affect the delivery of the multimedia data in both directions. When the operator carrying an MS walks around a metropolitan area, he/she may pass from a coverage area of one wireless network to another coverage area, causing a handoff process to be performed. In this case, some data frames the human operator collects with his/her MS may be lost in the communication with the remote CC, and a retransmission of those frames through a different wireless adapter is carried out by MEW. We have measured that the retransmission process delays the delivery of lost frames up to a couple of seconds, as indicated in the previously cited result (4). A couple of seconds can be an unacceptable delay in those cases in which a supervisor at the CC needs to promptly interact with a MEW mobile operator. Therefore, the one-way delivery delay exhibited by MEW should be kept within a few hundreds milliseconds. In particular, the ITU-T guidelines [13] recommend a one-way delay of up to 150 ms in order to use VoIP applications; however, for most practical purposes a limit of 300 ms can be acceptable.

In view of these observations, we have developed a solution that enables MEW to meet this requirement. Thus, the principal contributions of this paper can be summarized as follows.

This paper introduces an advanced end-to-end resource and mobility management mechanism over the standard TCP/IP protocol stack. This mechanism allows MEW to meet QoS requirements such as availability, continuity, and reliability of the communications, and interactivity in order to support effectively multimedia services such as IP telephony.

In addition, this mechanism enables the inter working of heterogeneous wireless technologies, including in the MEW MSs wireless adapters for broadband communications such as UMTS [22] or mobile WiMAX [12].

Specifically, this paper describes a session-layer load balancing mechanism that balances each single multimedia frame on the MSs' heterogeneous wireless adapters so as to maintain the audio/video data delivery time in both directions below the limit of 300 ms. In order to meet such QoS requirement, the load balancing uses an early retransmission technique that anticipates the retransmission of the suspected lost frames through a different wireless network adapter. In doing so, the technique exploits the overall bandwidth provided by the available heterogeneous MSs' adapters guaranteeing responsiveness and interactivity between the CC and the operators carrying the MSs, in spite of handoff and temporary congestion of some paths.

The remainder of this paper is structured as follows. The next section discusses the principal design issues of the MEW system. Section 3 describes the MEW resource and mobility management mechanism, whereas Sect. 4 presents our experimental evaluation. In Sect. 5, related works are compared and contrasted with our solution. Finally Sect. 6 concludes this paper.

2 Design Issues

Figure 1 illustrates the scenario in which the MEW system operates. There exist distributed Mobile Sensor (MS) devices worn by human operators (e.g., public officers), a remote Control Center (CC), and an infrastructure consisting of heterogeneous wireless and wired communication networks, available in an urban context, that allows the multimedia data exchange between MSs and the CC.

In this scenario, we have assumed the following.

In our current design, the CC is a remote storage server accessible neither to the mobile human operators nor to those involved in the recorded actions. At the CC resides a human operator (the so-called supervisor in Fig. 1) who may be responsible for sending control commands to the MSs or vocal information to the operators, once he/she receives multimedia data from the MSs. A centralized server such as that just introduced may be a bottleneck in our architecture if a large number of MSs transmit concurrently audio and video data streams to the CC. Nevertheless, our assumption does not prevent the CC to be designed as a redundant system consisting of a number of replicated CCs each of which is connected to a subsets of MSs (issues of design of a robust CC are beyond the scope of this paper).

The MEW system has been designed to deploy multi-homing facilities: the MS devices are equipped with heterogeneous wireless communication technologies. In particular, we assume that the MSs incorporate two or more wireless network adapters that offer connectivity on a local geographical scale and are conformant to the different IEEE standards such as 802.11b [10], and 802.11g [11], and one wireless network adapter offering connectivity on a large geographical scale such as UMTS [22] or mobile WiMAX [12]. In this latter case, we

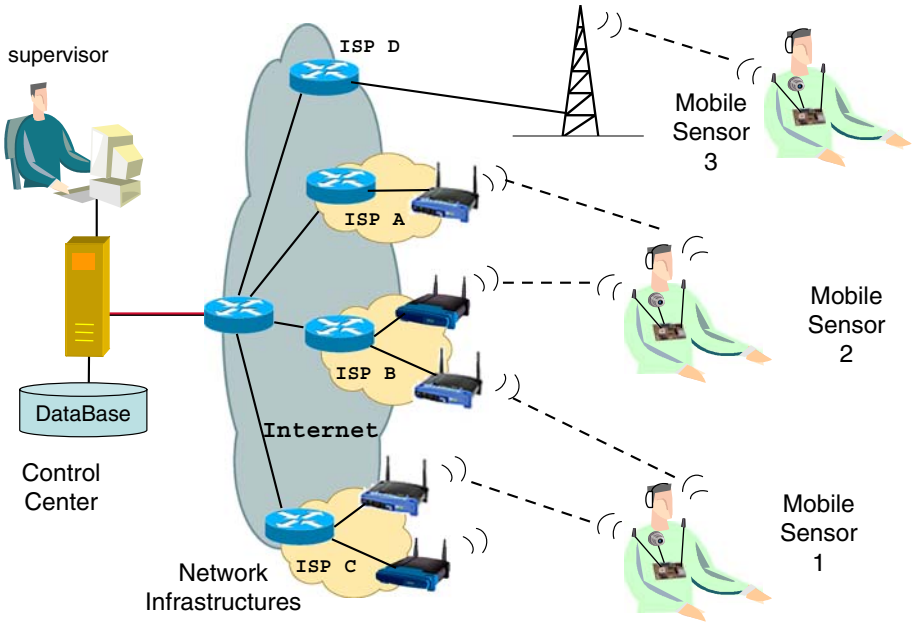

Fig. 1 Scenario

do not use GSM or GPRS communication infrastructures due to the insufficient bandwidth they may provide the MSs with for transmitting live video streams.

The MSs deploy a software architecture that incorporates a mobility and resource management mechanism. When an operator walks around the city, this mechanism dynamically selects the different access points and associates them with each MS wireless adapter in a way that is fully transparent to both the application and transport layers. The dynamic adapter selection and configuration allow the MSs to effectively use different heterogeneous network infrastructures to reach the remote CC, enabling availability and continuity of the communications. We assume the presence on the MSs of one wireless adapter for broadband communication technologies only, which is used when no connectivity on local scale is guaranteed. The mobile WiMAX or UMTS infrastructures require a larger energy consumption than that necessary to communicate over the IEEE standard technology (energy consumption is an issue as the higher the energy a wearable device consumes, the more severe the side effects it may have on the health of its bearer). Essentially, the medium-range technologies provide a bandwidth that is at least one order of magnitude higher than that provided by the long-distance ones and halve the power consumption in the time unit [7]. Hence, the assumption to use only one broadband communication technology when strictly necessary, thus preferring local range communication technologies, is motivated by our intent to limit the energy to which the MS human operator is exposed and to guarantee a longer MS battery duration.

Finally, we assume to support a variety of multimedia applications, ranging from video streaming to VoIP applications, over TCP/IP. In the scenario earlier described, a possible interaction between a supervisor at the CC and the mobile operators needs to be recorded in order to reconstruct, a posteriori, the precise sequence of the events. In particular, if a supervisor at the CC sends a critical vocal command to a mobile operator, it is crucial to assure that the voice communication is reliably delivered to the mobile operator. A UDP voice communication may not provide MEW with the required reliability guarantees.

3 MEW Mobility and Resource Management

We have designed and implemented an end-to-end advanced resource and mobility management mechanism to be incorporated in the MEW system, so as to allow MEW to operate successfully in the scenario previously described.

The mechanism is enabled by a MEW session-layer component termed *LoadBalancer*. The LoadBalancer component is deployed in the MSs and CC, and is responsible for balancing dynamically the multimedia load (i.e., the multimedia fragments obtained by the data frames of the application layer) among a number of SSL (over TCP) connections. The load balancing is based on an adaptive load balancing policy that monitors at run time connection QoS parameters such as the throughput and response time.

The mobility and resource management mechanism operates as follows.

The LoadBalancer component sets a 2-second timeout for a fragment it sends through a given SSL connection. This timeout is termed *Ack Timeout* (AT).

A large timeout value (2 s) allows the LoadBalancer not to close the SSL connection in case of network congestion, thus reducing the high delays that a new connection setup may require. Unfortunately, a large timeout value impedes MEW to timely retransmit fragments and entails delay peaks in the delivery of the fragments [7] and loss of interactivity.

To overcome this problem, our mechanism enables an *Early Retransmission* (ER) technique that consists of three algorithms (see next Subsections); namely, early detection of excessive fragment transmission delays, monitoring of the connection performance, and dynamic selection of a different and responsive connection through which transmit/retransmit a fragment. The aim of this technique is to limit the one-way delivery delay in both directions to few hundreds milliseconds. In other words, it aims at reducing and maintaining below the 300 ms the time elapsed from the creation of a fragment to be transmitted until that fragment is delivered to the destination. The limit of 300 ms [i.e., the so-called Maximum Allowed Delivery Time (MADT)] can provide MEW with an adequate level of interactivity for the majority of non real time applications.

In the design of the ER technique, we have made the assumptions described in the previous section and the following five hypotheses: (a) the maximum size of the fragment is 5,000 bytes, (b) in general, the wireless links are the bottlenecks and the main cause of packet loss in the paths between the MSs and the CC (c) the channels are approximately symmetric: the Round Trip Time can be considered the double of the latency, (d) the physical latency of the paths between the CC and a given MS does not exceed the limit of 80 ms (approximately a quarter of the MADT), which represents the typical latency between central Europe and the East coast of the US, and finally and most important (e) at each time instant, at least one of the paths between the CC and the MSs is able to deliver the fragment to the destination before a time limit of $(MADT/3)$ ms.

3.1 Early Detection of Excessive Fragment Transmission Delay

The ER technique uses an *Ack Warning Timeout (AWT)* that is set when the LoadBalancer sends a fragment. Its value is much lower than that of the AT timeout. If the AWT timeout expires before the acknowledgment for the fragment has been received, the SSL connection is declared "suspected" and the fragment is retransmitted through a different SSL connection (bound to a different network adapter), if available and sufficiently ready to transmit. The "suspected" SSL connection is however maintained open.

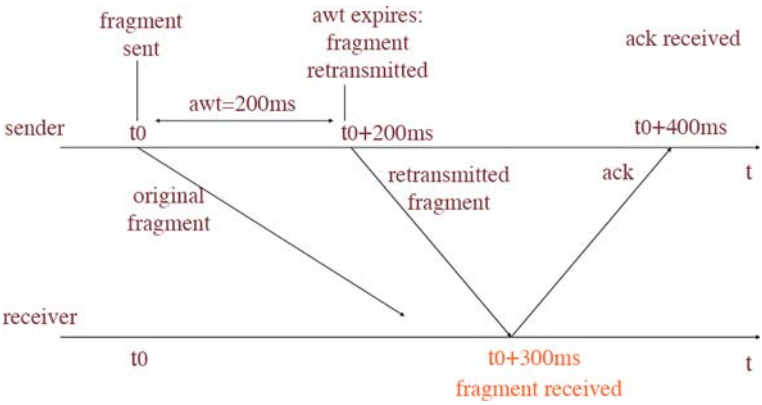

Fig. 2 Early retransmission scheme

With this approach, the fragment may reach the destination through two (or more) different paths. Each time the LoadBalancer at the destination receives a copy of the fragment, it sends an acknowledgment through the SSL connection from which the fragment has been received. The LoadBalancer uses a piggyback technique to aggregate more acknowledgments, if necessary, so as to reduce network utilization.

The AWT value choice represents one of the key aspects of the ER technique. In fact, we have assumed that the MADT is limited to 300 ms and that at least a path is "fast enough". Hence, the AWT value is set so that the "suspected" fragment is retransmitted within a time interval that allows it to reach the destination before 300 ms, once it has been produced by the communication sender. Moreover, the acknowledgment for the retransmitted fragment should be received before the AWT timeout for the retransmission expires. This allows the MEW load balancing mechanism to avoid a further (useless) retransmission. To this end, we have set the AWT equals to $2 * MADT/3$ (i.e., AWT is 200 ms).

As defined in our hypotheses, this value represents the double of the time before at least one of the available paths is able to deliver the fragment to the destination. Thus, let *t0* be the time instant at which a given fragment is produced by a sender and sent through a given path. If the fragment is not acknowledged before 200 ms after *t0*, a copy of the fragment is sent through a different path (the copy has a 200 ms limit before reaching the destination and delivering the acknowledgment, as depicted in Fig. 2). Assuming that the transmission delay of each path is identical in the two directions, if the acknowledgment for the retransmitted fragment copy is received before 400 ms after *t0*, we may conclude that the fragment has been received at the destination in time, i.e., before 300 ms after *t0*. Obviously, it is also possible that the original copy of the fragment reaches the destination before the retransmitted fragment copy is received.

Note that duplicated fragments are discarded by the receiving LoadBalancer component.

3.2 Monitoring the Connection Performance

The AWT value choice discussed earlier shows the following principal shortcoming: if the Round Trip Time of a given path exceeds the AWT value, the path is unable to carry the fragments and their acknowledgments within the limits, in milliseconds, that we have imposed. This leads the LoadBalancer to retransmit each fragment through that path.

In order to avoid an excessive and useless usage of the network bandwidth for retransmission purposes, the LoadBalancer monitors the performance of each path to detect those that are not sufficiently responsive (i.e., no good candidates for fragment transmission).

In particular, the LoadBalancer monitors the *Response Time* (RT), that is, the time elapsed from the transmission of a given fragment until the reception of its acknowledgment. Therefore, RT can be analytically described as in (1), where $T^{Ch}_{frag(i)}$ is the transmission time for a given fragment on the channel Ch and $T^{Ch}_{ACK_{frag(i)}}$ is the transmission time of the acknowledgment for that fragment on the channel Ch.

$$RT^{Ch}_{frag(i)} = T^{Ch}_{frag(i)} + T^{Ch}_{ACK_{frag(i)}} \tag{1}$$

The LoadBalancer declares "suspected" a connection that does not receive the acknowledgment before the AWT expires. In contrast, a "suspected" connection becomes "not suspected" when it receives an acknowledgment before the associated AWT expires. Moreover, when the LoadBalancer receives an acknowledgment for a given fragment, it firstly computes the RT and then the Response Data Rate (RDR). The RDR is the ratio between the size, in bytes, of the fragment (to which the LoadBalancer adds the size of the header) divided by the RT for the fragment, as described by (2) below.

$$RDR^{Ch} = \frac{size_{frag(i)} + size_{header\,frag(i)}}{RT^{Ch}_{frag(i)}} \tag{2}$$

For each channel, the LoadBalancer maintains the RDR of the last received acknowledgment in order to estimate the RT of each successive fragment that must be transmitted. Before a given connection receives the acknowledgment for the first transmitted fragment, the RDR of the connection is computed based on the elapsed time in the setup of the TCP connection itself (that is computed as the time consumed by the `connect()` socket system call) and on the size of the TCP segment that carries the SYN flag.

If at least one "not suspected" connection exists, a "suspected" connection cannot be selected as candidate for the transmission for a hibernating time period, which is twice the AWT timeout. After the hibernating time period, the LoadBalancer sends a light keepalive request (1 byte only to which the LoadBalancer adds the header) to the destination LoadBalancer. This latter LoadBalancer immediately replies with an acknowledgment in order to allow the sender to update the RT and the RDR of the "suspected" connection.

The same keepalive procedure as above is applied by the LoadBalancer to each "not suspected" connection that does not receive bytes for a period of 2 s (that is equal to the AT timeout period). This procedure allows the LoadBalancer to detect and close all those connections that are not functioning, even when no traffic is carried.

3.3 Selection of a Responsive Connection for Fragment Transmission

The LoadBalancer maintains a list of the fragments 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 connection. The following five principles guide the connection selection phase.

The first principle is based on the status of the TCP socket transmission buffer. In order to limit the communication delay in case a fragment is buffered for a large time period, it is important that the fragment is assigned to a connection "immediately ready to transmit" (i.e., a connection that has a TCP socket transmission buffer completely empty). However, the TCP performance may decrease if the LoadBalancer waits for the transmission buffer to

become empty before assigning a fragment to it; the TCP protocol cannot enlarge its congestion window and exploit the bandwidth available on the channel. The first principle represents a tradeoff between the two situations: a fragment is assigned to a connection "enough ready to transmit", that is, a connection whose TCP socket transmission buffer contains no more than 30,000 bytes.

The second principle concerns the constraint described above: the constraint is restricted when a candidate connection for the fragment transmission is suspected to be "not enough responsive". In this case, the fragment is assigned to the connection only when it is "immediately ready to transmit", that is, when its TCP socket transmission buffer does not contain any bytes and the fragment can be immediately transmitted to the destination.

The first two principles support the effectiveness of the third principle: more fragments are transmitted through the connections that appear more responsive. The faster a wireless adapter succeeds in delivering fragments, the sooner the TCP socket transmission buffer of the SSL connection used by that adapter becomes "enough ready to transmit" (and further fragments can be transmitted through that SSL connection).

However, the third principle does not prevent the transmission via the "not enough responsive" connections, as these connections eventually deliver their fragments and become "immediately ready to transmit". To overcome this effect, the fourth principle takes into account the recent history of the connections, by classifying them in "suspected" and "not suspected", as previously described. Thus, when a fragment needs to be delivered quickly (for instance, when the fragment has been transmitted and no acknowledgment has been received before the AWT) the LoadBalancer waits for selecting a "not suspected" connection that promises more responsiveness.

Finally, the fifth principle enables the use of broadband communication technologies only when all the local scale ones are not available (i.e., when the LoadBalancer is unable to detect the carriers of local scale access points).

Starting from these five principles, when the LoadBalancer needs to transmit a fragment, it selects a connection carrying out the following algorithm. The algorithm is illustrated, in form of pseudo-code, in Fig. 3 and described below.

If no local scale communication technologies are available, the LoadBalancer selects the broadband wireless communication adapter in order to send the multimedia fragment; in contrast, if local scale communications are detected, the algorithm operates as follows.

If all the available local scale connections are "suspected" the LoadBalancer selects those connections that are "immediately ready to transmit". If no connection is "immediately ready to transmit" the LoadBalancer delays the selection decision, waiting for a connection to become completely free. If more than one connection is "immediately ready to transmit" the LoadBalancer chooses the connection with the minor Estimated Response Time (ERT). This ERT represents the RT of a fragment as it was sent through a given connection at the current instant. Equation 3 shows the ERT of a fragment in a connection that is computed as the ratio between the sum of the fragment size, the LoadBalancer header size, and the number of bytes in the TCP socket connection buffer of the sender, divided by the RDR of the connection.

$$ERT^{Ch}_{frag(i)} = \frac{size_{frag(i)} + size_{header_{frag(i)}} + size^{Ch}_{TCPBuffer}}{RDR^{Ch}} \tag{3}$$

If there exists at least one "not suspected" connection, the LoadBalancer selects the "enough ready to transmit" and "not suspected" connections as candidate for the transmission. If more connections are "enough ready to transmit" and "not suspected", the LoadBalancer chooses the connection with the minor ERT. In contrast, if no "not suspected" connection

```
select connection for fragment transmission {

  if no carrier for short-distance communications adapters
     select long-distance connections
  else {
     if all connections are "suspected" {
       if ∃ "immediately ready to transmit" connections {
         select connection with lowest ERT
         return selected connection
       } else // delay selection decision
         return none
     } else // there are "not suspected" connections {
         if ∃ "enough ready to transmit" and "not suspected" connections {
          select connection with lowest ERT
          return selected connection
        } else {
          //no "enough ready to transmit" and "not suspected" connections
          if fragment has been previously transmitted {
             // delay selection decision
             // wait for a "not suspected connection"
             return none
          } else {
             // the fragment is to be transmitted for the first time
             if there are "immediately ready to transmit" connections {
               select connection with lowest ERT
               return selected connection
             } else // delay selection decision
               return none
          }
        }
     }
  }
}
```

Fig. 3 Connection selection algorithm in case of frame transmission

is "enough ready to transmit" the LoadBalancer behaves differently if the fragment is to be transmitted for the first time or it is to be retransmitted. If the fragment is to be retransmitted, the LoadBalancer delays the selection decision, waiting for a "not suspected" connection to become "enough ready to transmit". If the fragment has not been previously transmitted, the LoadBalancer candidates the "suspected" connections that are "immediately ready to transmit" and selects the connection with the lowest ERT. Finally, if no connection is "immediately ready to transmit" the LoadBalancer delays its selection decision.

Note that when all the transmission channels are busy in sending fragments to the destination, no further transmissions can be carried out. However, the LoadBalancer component is able to detect this condition and can be configured to raise an exception to the application level; then, this level is responsible for dealing with such an exception as this condition may lead to QoS requirement violations.

3.4 Implementation Notes

A prototype of the described MEW system has been implemented using a laptop equipped with the Linux operating system (Gentoo distribution) and the kernel version 2.6.21.

The prototype has been implemented so as to avoid busy waiting; to this end, the implementation is mainly based on a loop driven by the system call `select`. The `select` system call waits, without consuming CPU time, for some sockets to change their status, and then performs depending on the new socket state.

To the same aim, a list of the AWT timeouts for each sent packet is maintained; the `select` system call is invoked with the closest timeout in order to wake up the system when the timeout expires. In our implementation, the AWT has been set to 160 ms so as to take into account the delay that the operating system may introduce for both the process scheduling; in fact, the temporal precision of a general purpose Linux operating system is not less than 20 ms.

It is important to point out that, in order to decrease the time for the fragment delivery, the TCP sockets used for the transmission have been configured so as to disable the Nagle algorithm [18]. When few bytes are received from the application, this algorithm produces a delay before enabling the transmission in order to aggregate more bytes in a single TCP segment. The system call `setsockopt` with the parameters `IPPROTO_TCP` and `TCP_NODELAY` allows us to disable the Nagle algorithm for a socket, thus reducing the overall transmission delay.

4 Experimental Evaluation

We have carried out an experimental evaluation of the MEW prototype. The objectives and the principal results of this evaluation are summarized as follows:

1. The first objective was to assess the best set of parameter values for the retransmission algorithm. The results we have obtained in this case are quite surprising, especially for what concerns the definition of the "enough ready to transmit" connection;
2. The second objective was to assess the MEW ability to both effectively react to the changes in the availability of the communication resources, and deliver the multimedia frames to the destination within 300 ms. In this case, we have concluded that the network latency is a limiting factor, as it degrades the TCP protocol performance in case the protocol has to retransmit the fragments. In addition, as expected, in case of lost fragments, the protocols enabled at the different layers of abstraction (i.e., at TCP or application layers) are not capable of ensuring the fragment transmission within 300 ms if the channel latency is higher than 100 ms; this value represents the upper limit above which no guarantees on the frame delivery time can be ensured;
3. The third objective was to evaluate the influence of the frame size in transmitting the multimedia frames within 300 ms. In this case, if the size augments, the probability that the frames are lost and must be retransmitted augments correspondingly, thus introducing communication delays. Moreover, the TCP protocol introduces a delay for the retransmission that augments as the channel latency increases; hence, large frames are delivered to the destination "in time" (i.e., within 300 ms) as long as the channel latency is within 100 ms. We have been able to conclude that the best frame size is approximately 1 KB.
4. The fourth objective was to show the MEW ability to provide a high level of interactivity. In particular the principal result we have obtained in this case is that MEW can provide an end-to-end delay that is much lower than the mentioned ITU-T recommended delay for VoIP applications.

Finally we have measured the overhead introduced by our load balancing mechanism in terms of information we add to the frames in order to carry out the early retransmission algorithm. From the results we have obtained we can state that the overhead is negligible (less than 12 KB; for the sake of brevity, in this paper we do not report details of these results).

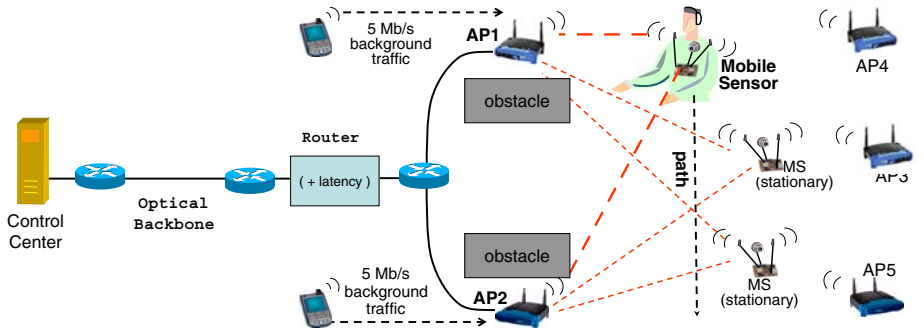

Fig. 4 Experimental scenario

4.1 Experimental Scenario

We have conducted the experimental evaluation in the urban area of our Department of Computer Science at the University of Bologna. A human operator, carrying an MS, has covered a route of 80 m within this area. The MS is equipped with two IEEE 802.11g wireless interfaces that can be associated with two access points available in the urban area.

The experimental scenario we have used is depicted in Fig. 4. As shown in Fig. 4, along this route there exist five IEEE 802.11g access points: two access points (i.e., AP1 and AP2 in Fig. 4) 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. The remaining three access points (i.e., AP3, AP4, and AP5 in Fig. 4) are not used by the MS and operate on channels that are adjacent to those on which the access points 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 −37 dBm; the power progressively decreases to −65 dBm in the middle of the path and to −90 dBm near the end of the path. In contrast, the access point AP2 works on the channel 6 and is located close to the end of the path the human operator covers. It is not accessible at the beginning of the path and it provides the MS with a power of approximately −92 dBm; this power becomes −77 dBm and then approximately −45 dBm at the end of the MS path (see Fig. 10 below). 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 access points are affected by an additional background traffic of approximately 5 Mbps. This traffic is generated by two mobile devices. These devices are not MEW components; rather they are mobile devices that operate in the urban area of our Department and are connected to the access point AP1 and AP2, respectively. In addition, there exist two stationary MSs that produce additional traffic on the same access points as those with which the MS in movement and the two aforementioned devices are associated.

The CC is located at the decentralized University of Bologna's departments in Cesena, approximately 80 km away from Bologna; the CC is connected to a network that consists of a 1,000 Mbps fast ethernet link and an optical fiber backbone connecting the University of Bologna central offices to its decentralized departments. The latency is approximately 4 ms.

A router is located at the beginning of the backbone. The router is a Linux machine that uses the iptables and netfilter services in order to intercept fragments that are sent from

and received by the MS. We have configured the router so that it buffers the fragments for a variable time, before transmitting the fragments to the destination. That buffering delay introduces a latency in the communications with the backbone of 50, 80, and 100 ms in the tests we have carried out (see below), respectively, with a tolerance of 7%. This choice allows us to analyze the MEW system behavior in a realistic scenario.

4.2 "Enough Ready to Transmit" Definition

Our first objective was to assess the best set of parameter values for the retransmission algorithm. To this end, we have run a set of tests in the scenario described above obtaining the following results, especially for what concerns the definition of the "enough ready to transmit" TCP connection.

We remind that a connection is "enough ready to transmit" if the TCP transmission buffer contains a number of bytes that is lower than a predefined N value. Our tests suggest that the N constant is equal to 30,000 B.

The N value should be sufficiently low in order to allow the LoadBalancer to assign a fragment to a connection capable of immediately transmitting it to the destination. In fact, a too high N may cause that a fragment is assigned to a connection that takes a long time before sending it, thus introducing a delay and augmenting the probability that the wireless channel becomes unusable for the presence of obstacles and inter-channel interferences. However, our tests show that a too low N value is not an appropriate choice. For example, if we set N equals to 5,000 bytes in a scenario with a latency of 50 ms, the average time necessary for transmitting frames of 1,126 B is three times higher than that we obtain with N equals to 30,000 B. This is due to the TCP behavior: when the TCP transmission buffer contains a few bytes (e.g., less than 5,000 B), these bytes could have been already sent to the destination but their acknowledgments could not have been received yet. Hence, waiting for the TCP to free its buffer before assigning it other data may lead TCP not to exploit the network bandwidth as it does not sufficiently open its congestion window.

4.3 Interactivity Analysis

The second objective was to assess the MEW ability to both effectively react to the changes in the availability of the communication resources, and deliver the multimedia frames to the destination within 300 ms. Note that the MEW system needs to send 25 frames each second to the destination. Each frame is 1,126 B.

This evaluation consisted of two principal phases: in the first phase, each access point was affected by a background traffic of 5 Mbps. In the second phase, we have injected, from a certain time instant, additional background traffic in the communication channel of the access point AP1 so as that access point was overloaded.

In both phases, we have evaluated the influence of the network latency using scenarios with latency of 50, 80, and 100 ms, respectively.

In the first phase, for each latency scenario, the frame transmission has been carried out in the following three modes: in the first two modes we have used the access point AP1 and access point AP2 in isolation, respectively, in order to show the performance of each single access point along the route of our experimental scenario. In the third mode, the MEW system exploits both the access points for the frame transmission. Figures 5 and 6 illustrate the frame delivery time we have obtained in case of latency scenarios of 50 and 80 ms, respectively.

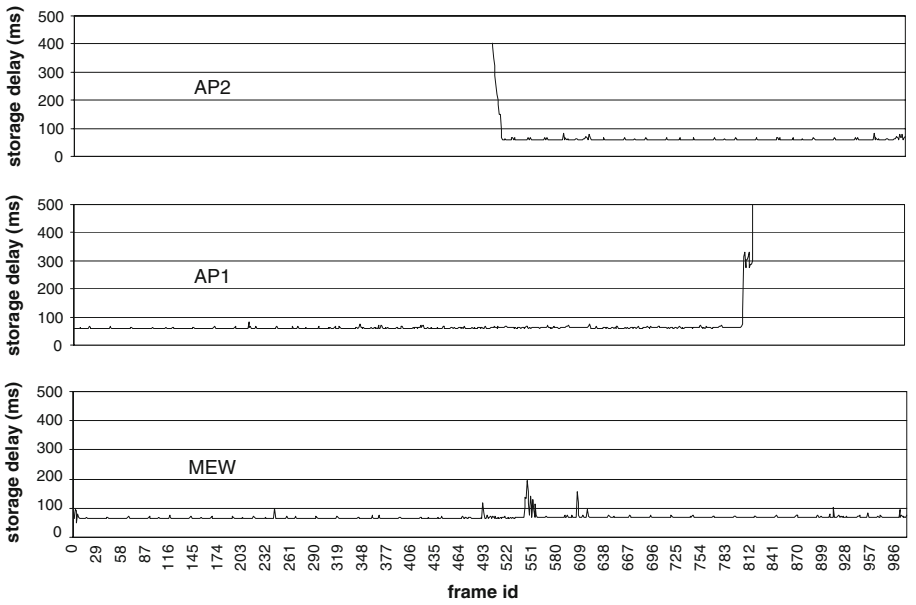

Fig. 5 Frames delivery time: 50 ms latency

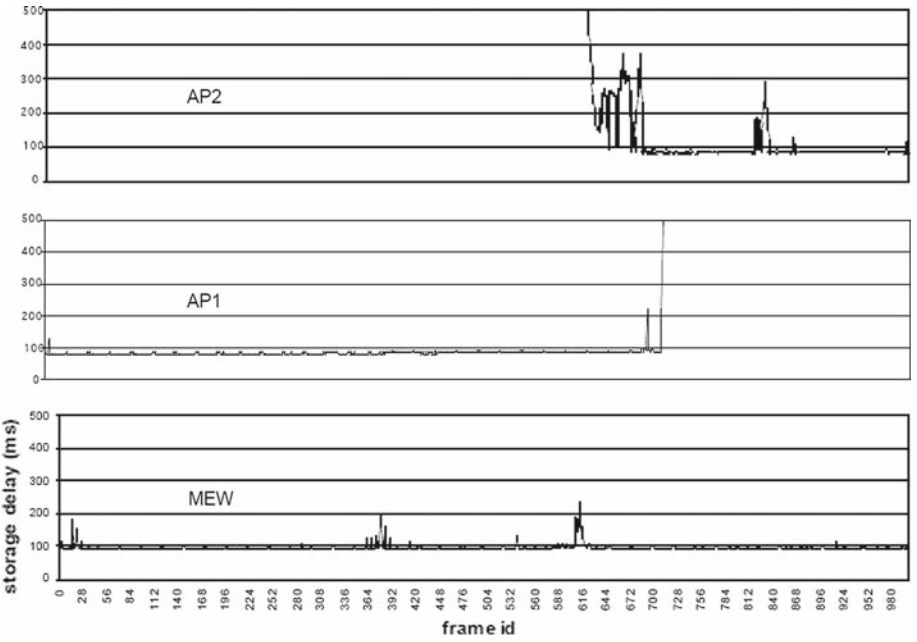

Fig. 6 Frames delivery time: 80 ms latency

In both scenarios, close to the end of the MS path, the access point AP1 firstly introduces considerable delays and then it becomes unavailable. The access point AP2 is unavailable at the beginning of the MS path and introduces small delays at the end of the route. The MEW

Table 1 Transmitted and retransmitted bytes through a different connection

		Latency 50 ms	Latency 80 ms	Latency 100 ms
AP1	Transmitted bytes	66,961 B	706,077 B	990,947 B
	Retransmitted bytes	9,096 B	12,507 B	642,939 B
AP2	Transmitted bytes	530,980 B	453,664 B	718,778 B
	Retransmitted bytes	17,055 B	11,370 B	400,515 B

system selects the suitable access point along the overall MS route, transmitting the frames to the destination within 300 ms.

In the scenario in which the network latency is 100 ms, the multimedia frames are delivered to the destination in time; however, their acknowledgments are not received within the timeout of 200 ms. Therefore, the majority of the sent fragments are retransmitted through different wireless network interfaces, causing a high network bandwidth usage, and consequently a high delay.

Table 1 shows the number of bytes that MEW sends with latencies of 50, 80, and 100 ms. As shown in this Table, when the latency is 50 and 80 ms, the amount of bytes retransmitted is small, i.e., approximately 23 and 21 frames, respectively, on a total of 1,000 frames. In contrast, with a high latency of 100 ms the percentage of retransmitted bytes is equivalent to the total number of frames.

This MEW behavior confirms that the interactivity is effectively supported, without waste of network bandwidth, in a scenario in which the latency does not exceed the 100 ms. With high level of latency the waste of bandwidth is significantly high.

In the second phase, the access point AP1 was overloaded due to additional background UDP traffic we have injected in the second part of the MS path, in order to carry out our experimental evaluation. The results we have obtained in this phase refer to the MEW behavior in case of latencies of 50 and 80 ms, and are shown in Figs. 7 and 8, respectively.

As shown in both Figures when the load of the communication channel of access point AP1 increases (the right axis of the Figures), we obtain a peak in the frame delivery time (storage delay in both Figures), that is approximately 220 ms in case of network latency of 50 ms, and approximately 270 ms in case of latency of 80 ms. The peaks are caused by the retransmissions of a few fragments using the slightly loaded access point AP2.

Although the presence of these peaks, it is worth noticing that MEW is capable of delivering fragments to the destination within our imposed limit of 300 ms.

4.4 Frame Size Analysis

The third objective was to evaluate the influence of the frame size in transmitting the multimedia frames within 300 ms. When the frame size augments, the number of IP datagrams necessary for the transmission increases accordingly, as the wireless channel MTU is sufficiently small (i.e., 1,500 B). The increasing number of datagrams to be transmitted leads to the following two effects.

Parts of the multimedia frames can be lost and must be retransmitted by TCP, thus introducing communication delays. In addition, the TCP retransmission phase is carried out when a timeout expires. This TCP timeout is dynamically sized accordingly to the estimated channel latency. Therefore, we conclude that using large frame sizes may cause unnecessary communication delays, especially in case of networks with high latencies.

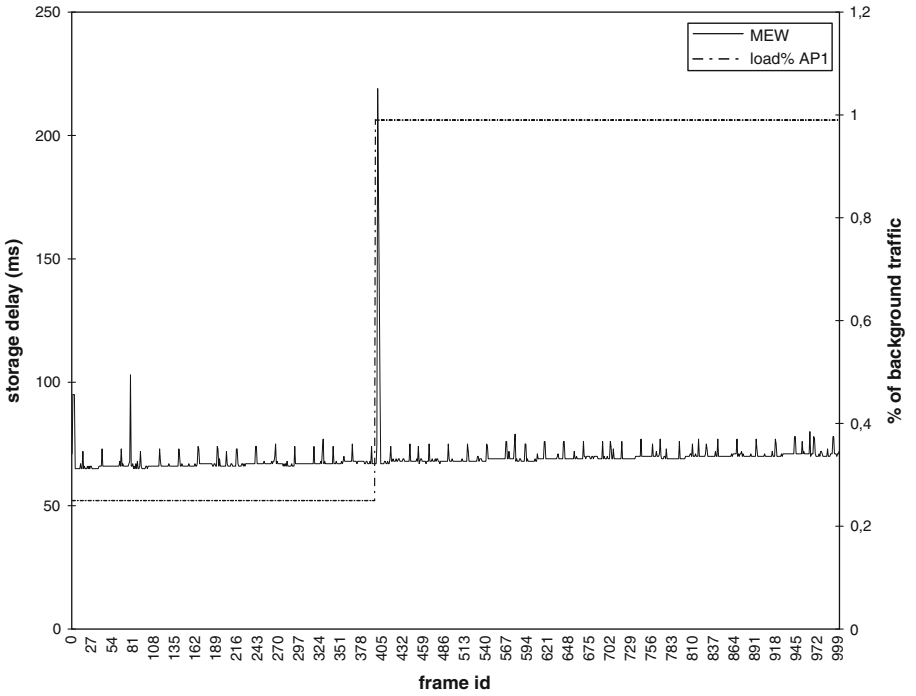

Fig. 7 Frames delivery time: 50 ms latency and additional load on AP1

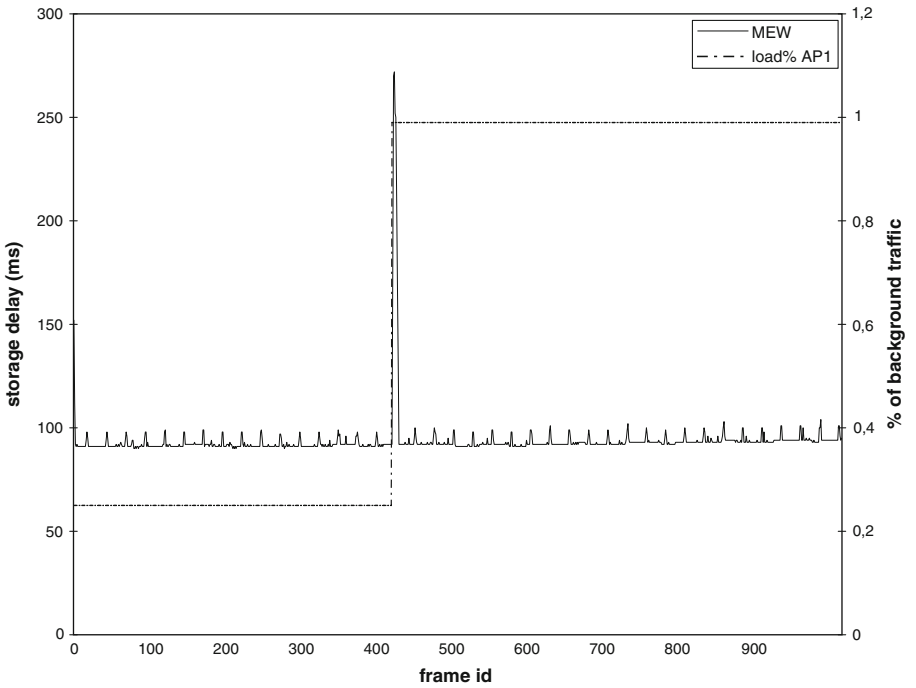

Fig. 8 Frames delivery time: 80 ms latency and additional load on AP1

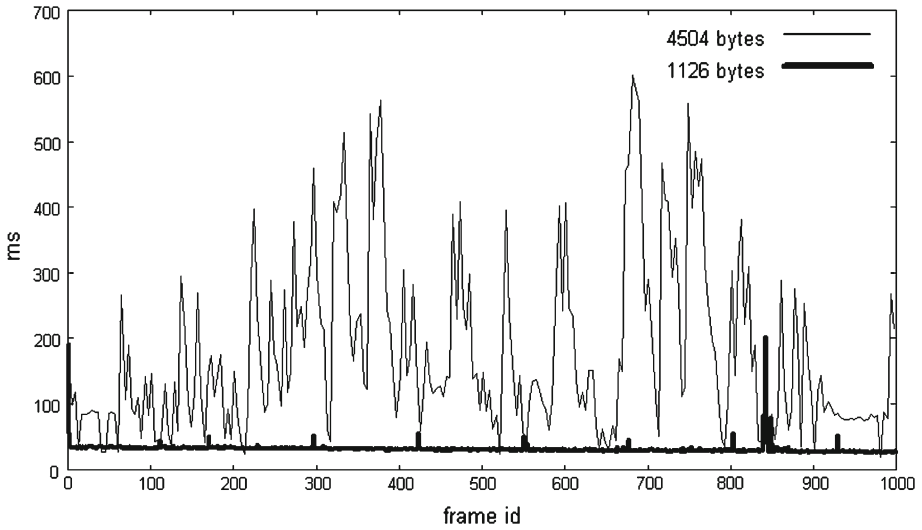

Fig. 9 "Frame storage delay" for frames of 1,126 and 4,504 bytes

The second effect is related to the TCP transmission with no fragment losses. The TCP protocol uses the "congestion avoidance" policy to limit, at the beginning of the communication, the amount of sent bytes, according to the "slow start" approach. This approach prevents the TCP to fully exploit the available network bandwidth. Hence, a large frame size is divided in successive segments that are sent introducing a small and undetectable delay.

Owing to these observations, we have carried out a set of tests when the network latency was fixed to 25 ms, with the purpose to assess the best frame size. Specifically, we have measured the frame delivery time in two different cases; with the first case we have sent 25 frames each second (each of which of 1,126 B). These frames are completely contained in the wireless channel MTU. In the second case, we have sent 6.25 frames each second, each of which of 4,504 B so that they must be transmitted in four IP datagrams. Note that the total number of transmitted bytes in the time unit is the same in both cases; hence, the different times we have obtained by these tests are due to the above mentioned effects, rather than the excessive load we have imposed in the second case with frames four times larger than those of the first case.

Figure 9 illustrates the results we have obtained. The first data series "1,126 bytes" in Fig. 9 represents the 1,000 frames of 1,126 B that have been sent by the MS according to the first case. These frames are delivered to the CC within 31 ms on average, with a maximum delay of 203 ms. Only two frames are retransmitted by the LoadBalancer; in the other cases, the losses are detected and resolved by the TCP transmission mechanisms.

In contrast, the data series "4,504 bytes" in Fig. 9 represents the 250 frames of 4,504 B that have been sent by the MS according to the second case. The frames are delivered to the CC within 177 ms on average, with a maximum delay of 601 ms. Moreover, the LoadBalancer has retransmitted approximately the 38% of the frames.

To conclude, these tests show that the frame size must be lower than the MTU of the wireless channel in order to avoid a high performance degradation. Specifically, in our scenario, the best choice concerning the frame size is 1,126 B.

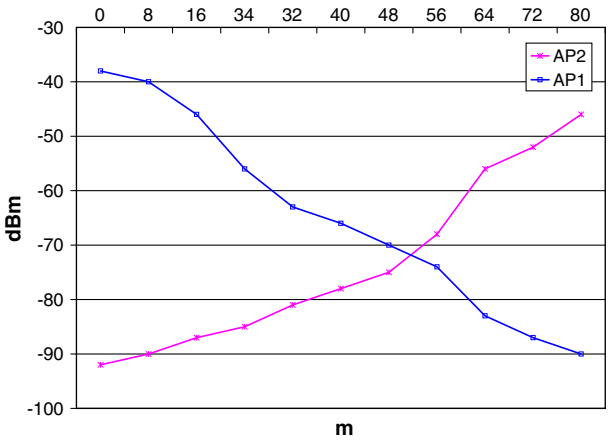

Fig. 10 Access point signal strength

4.5 Strict Responsiveness Analysis

Our last objective was to show the MEW ability to provide a high level of interactivity. Specifically, if the latency of at least one of the network paths does not exceed approximately 10 ms, MEW can support a MADT lower than 100 ms and then VoIP services with strict QoS requirements (such as those recommended by the ITU-T guidelines [13]).

A probing program has been written to simulate the traffic of a VoIP session, similarly to the approach adopted in [4]. The program sends a continuous sequence of frames through the two LoadBalancers from the CC to the MS. The payload size of each frames is 60 B and a frame is sent every 20 ms. This scenario simulates a VoIP session of 24 Kbps with no silence suppression. We recorded the delivery time of each frame from the CC to the MS.

The tests have been conducted in the experimental scenario previously described without introducing artificial latency: the backbone produces approximately 4 ms on average. In addition, we have modified the route covered by the MS with the aim to pass through three different coverage areas. Figure 10 shows the signal strength exhibited by the two access points AP1 and AP2, as it is perceived by the MS along the route. At the beginning of the route, the access point AP2 is unavailable for the MS as it provides the MS with a very weak signal strength, in the middle of the path both the access points provide the MS with a good signal strength, and at the end of the route the access point AP1 is no longer available.

In the tests, the frame transmission from the CC to the MS in movement has been carried out in three modes: the first two modes do not use the MEW system; they exploit a single TCP connection available through one of the access points in order to show the performance exhibited by that access point only. The third mode uses the MEW system that exploits both the access points. In Fig. 11, we compare and contrast the frame delivery time in these three modes. The first two graphs from the top of Fig. 11 represent the frame delivery time using the access point AP2 and access point AP1, in order. The bottom graph in Fig. 11 illustrates the MEW behavior.

When the access point AP1 is used in isolation, at the beginning of the route the MS delivers the frames to the destination with acceptable delays, in the middle of the route the MS exhibits delays up to approximately 416 ms, and at the end of the path it does not send any frame since AP1 results unavailable. In contrast, when the access point AP2 is used in

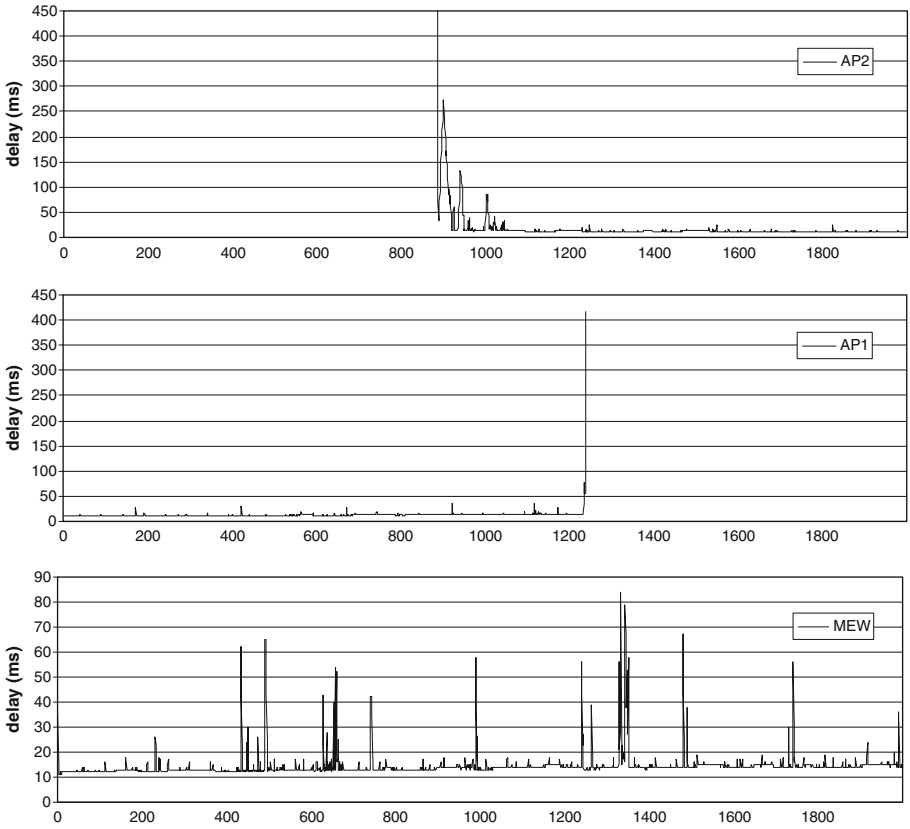

Fig. 11 Delivery time of VoIP frames

isolation, at the beginning of the route the MS cannot send any frame since AP2 is unavailable; in the middle of the route the access point AP2 becomes available and the MS that uses it for the frame transmission exhibits delays up to 1,319 ms. At the end of the path, the MS delivers the frames to the destination with acceptable delays.

In case both the access points are used in parallel by MEW, the frames are delivered to the destination with a maximum delay of approximately 83 ms as shown in the bottom graph of Fig. 11.

5 Related Work

The variety of wireless access technologies currently available are complementary when offering the user a set of differentiated services. It is quite clear that there will not be a single wireless access solution appropriate for all applications and scenarios. On the contrary, we expect that many wireless networks will coexist in heterogeneous contexts, allowing mobile and nomadic users to exploit the best available access service. The protocols for the integration, from the standpoint of the applications, of heterogeneous networks do not have a defined position in the classic Internet protocol stack, and are placed from the data link layer to the application layer, depending on the context.

However, a low-level approach needs to take into account and use the characteristics of each given wireless technology, and produce a solution that is not independent of the underlying layers. Other approaches [5,6] suggest placing the resource and mobility management at the network and transport layers, respectively. Specifically, as for the transport layer approach, current state of the art has produced a number of research works and protocols. These protocols can be classified into two principal categories; namely, the protocols that propose enhancements to the TCP protocol and those that can be considered as TCP substitutes.

The former category includes protocols such as pTCP [9], R-MTP [17], mTCP [24] that transmit each packet on the best available wireless interface; however, they do not investigate issues of dynamic configuration and run time reconfiguration of heterogeneous wireless network interfaces, as we do in our MEW system.

The latter category includes such protocols as SCTP [19,21] and the mobile extension of SCTP termed mSCTP [14,23].

SCTP (Stream Control Transmission Protocol) [19,21] is a standardized general-purpose transport protocol that can be effectively used to support message-oriented applications, and is built on top of the IP protocol. SCTP has been designed to replace the standard TCP protocol in typical multimedia contexts in which the mechanisms that TCP employs can result too restrictive. SCTP shares a number of similarities with TCP: it is connection-oriented and reliable. Specifically, the SCTP reliability is guaranteed by acknowledging the data that are transmitted: if a data is not acknowledged, it is retransmitted. However, SCTP differs from TCP for the following two principal features: (1) SCTP divides up the data to be transmitted into different streams. Each stream is independent from the other streams and is further segmented in chucks that are typically acknowledged using selective acknowledgements; (2) SCTP interacts with the IP layer by allowing an association (i.e., a connection established between two end-points) to use a range of available IP addresses.

This latter SCTP feature suggests that the protocol is capable of providing the upper layers with multi-homing capabilities. However, SCTP does not use multiple links for load balancing, as data transmission on multiple paths may cause packet reordering that leads to congestion control issues (SCTP is based on the TCP congestion control mechanism that does not support multihoming). Rather, SCTP uses one of the available addresses as primary address for the transmission (i.e., primary path) and the others as secondary addresses: in case an error occurs at run time, SCTP retransmits the chucks data to a secondary address. All Retransmission to Alternate is the retransmission policy used by SCTP. This policy deals with network congestion and path failures by sending all retransmission to an alternate secondary IP address. However, it has been shown that the SCTP's retransmission policy exhibits performance degradation. In addition, the addresses used for multi-homing purposes are fixed and known in advance, thus preventing the use of SCTP in a mobile environment where local addresses of mobile devices constantly change as effect of mobile device movements. To overcome this latter SCTP limitation a further protocol termed mSCTP (mobile SCTP) [14,23] has been developed.

In mSCTP a mobile host can own (at least) two IP addresses in a SCTP association during the handoff phase, if two access points are simultaneously available; mSCTP exploits the overlapping of coverage areas, that is, the coverage area of an old access point to which a mobile device is connected and that of a new access point to which the mobile device will perform a handoff. Hence, the mobile host can obtain the IP address from the new access point and use it to prepare the handoff process, i.e., to modify the set of IP addresses that describe the SCTP association.

The protocols we have summarized so far share a number of similarities with our load balancing mechanism. As in our case, all of them use the estimation of the available bandwidth and timeout techniques to transmit packets to a different wireless path in order to guarantee availability, reliability, and continuity of the communications. In particular, we might state that the emerging mSCTP protocol can be viewed as a direct antagonist of our solution. However, the solution we have deployed in the MEW mobile surveillance system is implemented at a higher layer of abstraction (at the session-layer) and in addition to the load balancing among available paths, it is capable of carrying out a dynamic and automatic configuration of MS heterogeneous wireless interfaces, and enabling the routing of each multimedia data packet.

In addition, all above protocols route flows of packets to an alternative path, when they detect congestion or failures of the path used for the communication. In contrast, our approach is fine-grained and more responsive to the network condition changes as we select the path that each single packet (rather than flows of packets) has to follow in the communication with the remote CC.

In essence, whereas at network and transport layers the research activity has been very active in the last years, it is not equally true for session layer approaches. In fact, there exist very few works, available in the literature, that focus on session layer handover and resource and mobility management mechanisms. These works, which include [1] and [3], propose middleware solutions based on the use of proxies to support handoff.

Landfeldt et al.[15] presents a framework termed SLM that extends the TCP/IP model by introducing a session layer that switches TCP streams between connections. Although our solution is applied at the same abstraction layer as that of SLM, our session layer LoadBalancer component uses simultaneously all the available connections in order to transmit the multimedia frames to the CC.

ABC techniques [8] exploit multiple access networks available on board of a variety of mobile devices. The concept behind ABC is to allow users not only to be always connected, but to be also connected to the best available device and access technology at all times.

The ABC approach shares the same objectives as ours, i.e., guaranteeing continuity and availability of the communications. However, in ABC the best access network or device among those available is chosen for data transmission; in contrast, we use concurrently all available heterogeneous access networks by dynamically balancing each packet among them based on such QoS requirements as the throughput and delay.

In the near future, 4G mobile communications should converge the advanced wireless mobile communications and high-speed wireless access systems into an Open Wireless Architecture platform (OWA) [16], which becomes the core of this emerging next-generation mobile technology. OWA defines the open interfaces in wireless networks and systems, including base-band signal processing parts, Radio Frequency (RF) parts, networking parts, and OS and application parts, so that the system can support different industrial standards and integrate the various wireless networks into an open broadband platform.

Unfortunately, currently these solutions are not completely developed. In addition, the existing ones do not cross the domain boundaries. Thus, from our point of view, a better solution for the described scenario is to locate the responsibility of the wireless communication integration in a separate session layer, that is fully independent of the communication technologies.

Finally, as for application level load balancing techniques, most of the publications in the public literature focus on mechanisms applied in the wired/Internet environments, in the context of Web applications, as documented in [2]. To the best of our knowledge, load balancing

and multihoming in the context of wireless and mobile environments are mostly enabled at the transport layer, as previously described.

6 Concluding Remarks

We have described the design and experimental evaluation of an extension of the MEW surveillance system. This extension allows MEW to both use broadband and local scale wireless communication technologies, and effectively meet interactivity requirements with the purpose of supporting VoIP multimedia services.

In particular, the principal objective of the MEW system we propose in this paper is to exploit the heterogeneity of the communications in order to develop an end-to-end resource management mechanism that masks to the applications the presence of different underlying wireless network infrastructures. In MEW, the transmission channel is characterized in terms of throughput and latency that are continuously monitored so as to balance the multimedia load among different wireless network adapters. This solution allows us to address issues of communication resource fluctuation.

Our experimental evaluation has shown that MEW is able to support the transmission of multimedia data frames of 1,126 B, and the interactivity, without waste of network bandwidth, in a real experimental scenario in which the latency does not exceed the 100 ms. In addition, MEW guarantees a frame delivery time lower than 100 ms in a scenario that used an emulated VoIP application. Hence, VoIP services with strict QoS requirements (such as those recommended by the ITU-T guidelines [13]) can be used by MEW in case a supervisor at the CC must promptly interact with a human mobile operator.

Our future works include the design of a fully decentralized CC architecture in order to enable the MEW system to scale out and accommodate an arbitrary large number of MSs, and the evaluation of the overall MEW system using the popular NS-2 network simulation tool [20]. In addition, we are planning to develop an analytical model of our end-to-end resource and mobility management mechanism.

References

1. Bellavista, P., Corradi, A., & Foschini, L. (2005). Application-level middleware to proactively manage handoff in wireless internet multimedia. In *LNCS 3754 Management of Multimedia Networks and Services*, 17 October 2005.
2. Cardellini, V., Casalicchio, E., Colajanni, M., & Yu, S. P (2002). The state of the art in locally distributed web-server systems. *ACM Computing Survey, 34*(2), 263–311.
3. Chan, J., De Silva, R., Zhou, S., & Seneviratne, A. (1998). A framework for mobile wireless networks with an adaptive QoS capability. In *Proceedings of the 5th International Workshop on Moble Multimedia Communication MoMuc '98*, October 1998.
4. da Conceicao, A., Jin, L., Florencio, D. A., & Kon, F. (2006). Is IEEE 802.11 ready for VoIP? In *Proceedings of 8th Workshop on Multimedia Signal Processing*, October 2006.
5. Eddy, W. M. (2004). At what layer does mobility belong? *IEEE Communications Magazine, 42*(10), 155–159.
6. Fodor, G., Eriksson, A., & Tuoriniemi, A. (2003). Providing quality of service in always best connected networks. *IEEE Communications Magazine, 41*(7), 154–163.
7. Ghini, V., Lodi, G., & Panzieri, F. (2008). Mobile E-witness. *Multimedia Tools and Applications, 37*(3), 293–318.
8. Gustafsson, E., & Jonsson, A. (2003). Always best connected. *IEEE Wireless Communications, 11*(1), 49–55.

9. Hsieh, H. Y., & Sivakumar, R. (2002). pTCP: An end-to-end transport layer protocol for striped connections. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP02)*, 2002.
10. IEEE Std 802.11b-1999. (1999). Higher speed physical layer (PHY) extension in the 2.4 GHz band. *IEEE Standard for Information Technology*.
11. IEEE Std 802.11g-2003. (2003). Further higher-speed physical layer extension in the 2.4 GHz band. *IEEE Standard for Information Technology*.
12. IEEE Std 802.16e-2005. (2005). Physical and medium access control layers for combined fixed and mobile operation in licensed bands. In *Amendment to IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, 2005.
13. ITU-T Recommendation G.114. (2003). One-way transmission time, May 2003.
14. Koh, S. J., Chang, M. J., & Lee, M. (2004). mSCTP for soft handover in transport layer. *IEEE Communications Letters, 8*(3), 189–191.
15. Landfeldt, B., Larsson, T., Ismailov, Y., & Seneviratne, A. (1999). SLM, a framework for session layer mobility management. In *Proceedings of the 8th International Conference on Computer Communications and Networks*, October 1999.
16. Lu, W. W., Miao, K., Zhang, P., & Maes, S. H. (2007). Technologies on the future converged wireless and mobility platform. *Guest Editorial in IEEE Wireless Communications Magazine*, April 2007.
17. Magalhaes, L., & Kravets, R. (2001). Transport level mechanisms for bandwidth aggregation on mobile hosts. In *Proceedings of 9th International Conference on Network Protocols (ICNP01)*.
18. Nagle, J. (1984). Congestion control in IP/TCP internetworks. IETF Request For Comments 896, January 1984.
19. Noonan, J., Perry, P., & Murphy, J. (2002). A study of SCTP services in a Mobile-IP Network. In *IT&T Annual Conference*, October 2002, WIT, Ireland.
20. NS-2 simulation tool. (2008). Retrieved from http://nsnam.isi.edu/nsnam/index.php/Main_Page, 2008.
21. Stewart, R., Xiei, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., et al. (2000). Stream control protocol. In *RFC 2960*, October 2000.
22. UMTS Project home page. (2007). Retrieved from http://www.umts-forum.org/, 2007.
23. Xing, W., Karl, H., & Wolisz, A. (2002). M-SCTP: Design and prototypical implementation of an end-to-end mobility concept. In *Proceedings of 5th International Workshop the Internet Challenge: Technology and Applications*, October 2002, Berlin, Germany.
24. Zhang, M., Lai, J., Krishnamurthy, A., Peterson, L., & Wang, R. (2004). A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *Proceedings of the USENIX 2004 Annual Technical Conference*, June 2004.

Author Biographies

Vittorio Ghini is a research associate at the Computer Science Department of the University of Bologna (Italy) since December 2003. He received the "Laurea" (1997) and Ph.D. degrees (2002) in Computer Science from the University of Bologna. His research interests include distributed multimedia systems, middleware protocols for QoS over IP networks, and dynamic multihoming 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. She is currently a research associate of Computer Science at the University of Bologna (Italy). Her research interests include distributed systems, middleware, application server technologies, clustering techniques and SLA management.

Stefano Cacciaguerra is a Technologist at the Department of Bologna, INGV. He received the Laurea degree in computer science in 2001, and the Ph.D. degree in 2005, both from the University of Bologna, Italy. His research interests include wireless networks, distributed systems, Internet games, and multimedia applications.

Fabio Panzieri received the "Laurea" degree in "Scienze dell' Informazione" from the University of Pisa (Italy), and the Ph.D. degree in Computer Science from the University of Newcastle upon Tyne (UK). He was appointed a professor of Computer Science at the University of Bologna (Italy) in 1990. His research interests span many areas of distributed computing, including fault-tolerance, real-time, and middleware and communication support for large scale distributed applications.