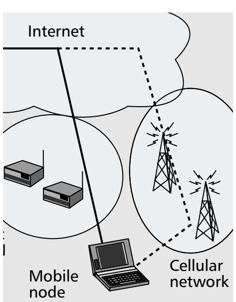
SEAMLESS INTERNETWORKING OF WLANS AND CELLULAR NETWORKS: ARCHITECTURE AND PERFORMANCE ISSUES IN A MOBILE IPV6 SCENARIO

MASSIMO BERNASCHI, ITALIAN NATIONAL RESEARCH COUNCIL
FILIPPO CACACE, GIULIO IANNELLO, AND STEFANO ZA, UNIVERSITY CAMPUS BIO-MEDICO
ANTONIO PESCAPÈ, UNIVERSITY FEDERICO II OF NAPLES



The authors review the problem of network mobility and internetworking between heterogeneous data networks and present an approach to the integration of WLAN and cellular networks based on loose coupling and the use of emerging mobility protocols.

ABSTRACT

We review the problem of network mobility and internetworking between heterogeneous data networks and present an approach to the integration of WLAN and cellular networks based on loose coupling and the use of emerging mobility protocols. The handoff performance of such an approach is studied, at the network and transport levels, in a realistic scenario along with the impact on global performance of transport protocols. Finally, a method of eliminating any packet loss at the network layer during handoff is presented and evaluated.

INTRODUCTION

The integration of wireless LAN (WLAN) based on IEEE 802.11 and cellular networks offers several advantages. Although users take advantage of high-speed data connections only in proximity of hot spots they can, thanks to the geographical coverage of cellular networks, maintain seamless connectivity, albeit at reduced speed. Such integration, however, requires the solution of problems at different levels: from the definition of an internetworking architecture to the management of user authentication and billing.

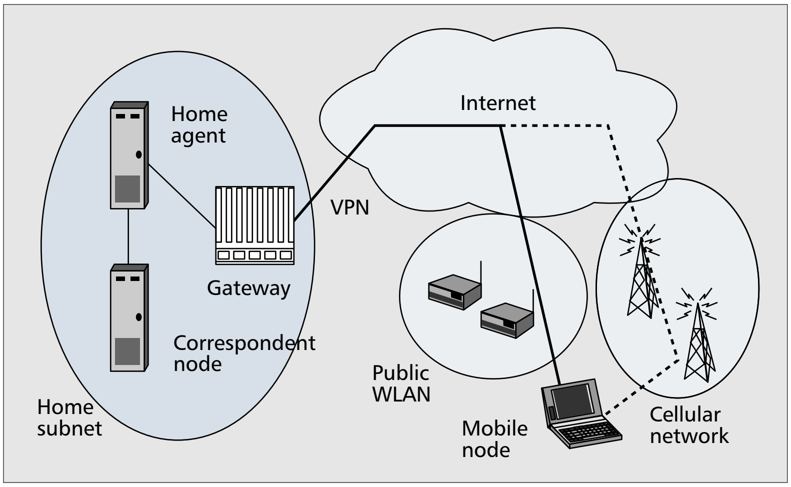
In this article we present an approach to the integration of WLAN and cellular networks based on loose coupling and the use of emerging mobility protocols that is very flexible and simple to implement. It requires only the introduction of a specialized mobility protocol in the network stack of the mobile host with no additional requirement to existing wireless networks or the Internet infrastructure. Since there are some concerns about its efficiency, we decided to characterize the performance that can be expected, not only in theory or by means of simulations, but in a real usage scenario. Our study is not limited to the performance of the network layer, even if this is crucial to determine the *handoff delay*; it also considers the transport protocols, since these determine the overall performance of network applications (e.g., file transfer, multimedia streaming, voice and video communications), and, as a consequence, the final user perception of network usability

The remainder of this article is organized as follows. We introduce the terminology and solutions to network mobility problems, and describe a possible WLAN/cellular internetworking scenario. We then present both a model and experimental results on internetworking performance at the network layer. We discuss how to improve this performance. We devote a section to evaluating performance at the application layer. We then offer some concluding remarks.

BACKGROUND ON NETWORK MOBILITY

The solutions proposed for the integration of WLAN and cellular networks are usually classified as tightly coupled or loosely coupled [1]. In the tightly coupled approach the WLAN appears to the cellular core network as another cellular access network. The WLAN emulates functions that are natively available in cellular radio access networks, so the WLAN network cards need to implement the cellular protocol stack. In loosely coupled solutions, WLAN and cellular networks are completely separated and only connected through the Internet. The mobile node (MN) must resort to a mobility protocol like Mobile IP to manage the internetworking between WLAN and cellular networks. This implies the availability of a mobility gateway on the MN's home subnet known as a *home agent* (HA). Other specific mechanisms must be sup-

 1536-1284/05/$20.00



■ **Figure 1**. *The mobility scenario.*

plied if authentication, billing, and QoS guarantees need to be managed seamlessly between the two access networks. Even if no generally accepted conclusion has been reached yet, there is some evidence that loosely coupled approaches offer several architectural advantages, with no evident drawbacks [2]. Hereafter, we only consider the loosely coupled approach based on the Mobile IPv6 protocol.

Mobile IPv6 [3] is designed to support mobile nodes in IPv6 networks. When an MN moves through other networks, the HA keeps track of the current binding of the MN. When a handoff takes place, the MN sends messages, known as binding updates (BUs), to both the HA and any node with which it is communicating, usually indicated as *correspondent nodes* (CNs). If a CN does not support Mobile IPv6, packets are sent through the HA by means of IPv6 tunneling between the HA and the MN, resulting in triangular routing like in Mobile IPv4. If the CN understands BUs it may bypass the HA and route its own packets directly to the MN by using a special option in the IPv6 routing header. The above mechanism is called *horizontal handoff* if the migration is between homogeneous networks. The same procedure can also be used for mobility through heterogeneous networks for MNs equipped with more than one network interface. In this case the change of active interface is called *vertical handoff*, since it takes place in the presence of a hierarchy of overlaid networks with different features (e.g., bandwidth, power consumption, cost).

Performance of Mobile IPv6 with respect to horizontal handoffs has been studied thoroughly in the last years [4, 5]. Several mechanisms have been proposed to enhance and optimize aspects of the protocol, in particular to reduce handoff delay and packet loss. Even though vertical handoffs can be implemented through Mobile IP, their peculiarities justify specific analysis [6, 7]. Foremost, horizontal handoffs are typically required when an access router becomes unavailable due to mobile host movement. In other words, multiple access routers are usually not accessible at the same time for long periods. On the contrary, being overlaid is a common situation for heterogeneous networks. An implication of the above observation is that handoff can be initiated for convenience, rather than connectivity, reasons. Moreover, handoff latency and packet loss are affected by network overlay. It is often possible to have lossless handoffs by performing all the configuration and signaling steps on the new network before actually leaving the old one. This is considered a *soft handoff* as opposed to a *hard handoff*, which happens when the active access router becomes unreachable before handoff execution.

Figure 1 shows the mobile scenario we consider (for similar usage scenarios see [8]). An MN has seamless access to the private intranet and public Internet through multiple network interfaces such as Ethernet cards, wireless cards and 3G data cards. When connected through public hotspots or cellular networks, the MN resorts to a virtual private network (VPN) to create a secure connection to the corporate intranet. Authentication and security issues are handled by the VPN application, and we do not consider these issues in the sequel. The corporate gateway, which is the arrival point of the VPN, is configured to see the VPN as an additional IPv6 subnet, through which it sends router advertisements that are used by the MN to configure its care-of address away from home. We also assume that the intranet is IPv6-enabled, and that both the MN and CN are IPv6 hosts. Actually, the testbed we implemented in order to evaluate the performance of this configuration includes mechanisms to cope with IPv4 hosts. However, due to lack of space, these are not discussed here.

Internetworking Performance at the Network Layer

Two issues must be addressed in order to achieve seamless internetworking:

- Established connections should not be broken when the mobile host changes the active network interface.
- The service disruption should be so short that it does not preclude satisfactory use of network services.

As described earlier, the transparency to existing connections is obtained at the network (i.e., IP) layer. As to the second point, in general, the handoff process can be seen as being composed of two phases:

- Handoff detection
- Handoff execution

The optimization of handoff performance should attempt to shorten both phases. However, the detection phase is particularly important when moving through heterogeneous networks, for example, when a mobile host is connected to a corporate WLAN and is also under the coverage of a cellular network. A delay on the order of seconds to disconnect from the WLAN and transfer its connections to the GPRS/UMTS card is hardly acceptable in this situation, since the mobile host is already connected through both interfaces. Vertical handoffs can be divided in:

- *Forced handoffs*, triggered by physical events related to network interface operability issues,

as in the case described above of a mobile host losing its connection to the wireless access point. In our context, forced handoffs are usually from WLAN to cellular networks.
- *User handoffs*, triggered by user policies and preferences. The typical situation is when the mobile host discovers the availability of a faster or cheaper network. In our context, this means a handoff from cellular networks to WLAN.

The fundamental difference between the two kinds of handoffs is that in the first case the preferred network is no longer available after the handoff, whereas in the second case it is available only after the handoff. Detection and triggering are more important for forced handoff, since the service disruption is proportional to the delay after which the mobile host becomes aware of being disconnected from the preferred network. In user handoffs, the mobile host could postpone handoff execution until the newly selected interface is properly configured. Moreover, simultaneous use of the old and new interfaces could allow lossless handoffs. The handoff latency at the network level is due to the combined effect of:
- Delay D_t for detecting physical events eventually leading to the handoff
- Delay D_n for detecting the new access router and configuring an IP address on the new subnet
- Delay D_s for handoff execution, including delay for selecting the new router, sending signals to HA and CN, and latency before the arrival of packets on the new subnet

Note that the handoff execution always follows the first two phases, but phase 1 and 2 can overlap or even happen in reverse order, since all network interfaces could be up and configured before the handoff starts. Deeper insight on these three parameters is convenient for our subsequent analysis of handoff delays.

Current specifications of Mobile IPv6 rely on router advertisement (RA) messages received from routers for detecting mobility events (*L3 detection*). When a timeout for the maximum RA advertisement interval (T_{RM}) from the current router expires, the mobile host can execute the neighbor unreachability detection (NUD) procedure. Thus, D_t delay depends on T_{RM} and NUD delay, whose duration may vary from about 0.2 s to more than 4 s, and may have a relevant impact on the overall handoff delay. Note that for forced handoffs the NUD procedure is necessary, since only the unreachability of a higher-preference network can force the handoff to a lower-preference interface. For a user handoff, NUD is not necessary. The maximum RA interval is critical for layer 3 (L3) detection, and the most recent Mobile IPv6 specification indicates a maximum value of 0.07 s for access routers supporting mobile hosts. However, it is not reasonable to demand such a high frequency on cellular networks, where it would consume the scarce bandwidth. Moreover, packet buffering in these networks would nullify the high frequency of RAs, since a group of them would arrive at the same time to the mobile host.

Configuration delay D_n is the time required to configure an address on the new subnet and is mainly due to the duplicate address detection (DAD) procedure. For vertical handoffs D_n does not contribute to handoff delay since DAD is performed before the handoff. Moreover, Mobile IPv6 implementations usually do not wait for the DAD termination before using a new address.

	User handoff	Forced handoff
From WLAN to GPRS	1187 ± 340	2552 ± 433
From WLAN to UMTS	637 ± 120	1657 ± 272
Packet loss	no	yes
Delay depends on	T_{RA}, cellular RTT	T_{RM}, WLAN NUD, cellular RTT
From GPRS to WLAN	393 ± 8	1960 ± 210
From UMTS to WLAN	393 ± 8	1640 ± 130
Packet loss	no	yes
Delay depends on	T_{RA}, WLAN RTT	T_{RM} , cellular NUD, WLAN RTT

Table 1. *Network layer handoff delays (in milliseconds).*

Signaling delay D_s is the delay for handoff execution. It is the time interval between the sending of the BU to the HA and the arrival of the binding acknowledgment on the new interface. D_s is influenced only by the round-trip time (RTT) between these two nodes and the load on the network. Typical values for D_s range from 6 ms for WLANs to 800 ms for GPRS links and 250 ms for UMTS links.

Table 1 summarizes some experimental results collected in our testbed. The RA frequency on both networks is set to typical values for a mobile access router: T_{RM} is equal to 1500 ms and the average RA interval (T_{RA}) is equal to 775 ms. The NUD delay is approximately 650 ms on WLAN, 1200 ms on GPRS, and 900 ms on UMTS. Experimental tests were performed on Linux 2.4.22 Pentium IV PCs, using an MIPL 1.0 Mobile IPv6 implementation (http://www.mipl.mediapoli.com/). Each test was repeated 10 times by using dedicated wireless network access points, with no other traffic on them.

In a real scenario the two most common cases are user handoff from cellular network to WLAN and forced handoff from WLAN to cellular network. Since there is no packet loss, the delay is not relevant in the second case and there should be no service disruption at the network layer (even if at the transport layer the effect is different as we shall see later). In the first case, however, there is significant packet loss. The delay is composed of three parts: the delay for handoff triggering ($T_{RM}/2 = 750$ ms), the NUD delay (650 ms), and the signaling delay D_s (about 800 ms for General Packet Radio Service, GPRS, and 250 ms for Universal Mobile Telecommunications System, UMTS).

Improving Performance at the Network Layer

The latency of the handoff mechanism is the sum of the latencies of the detection and execution phases. The results of the previous section show that the detection delay D_t is a major com-

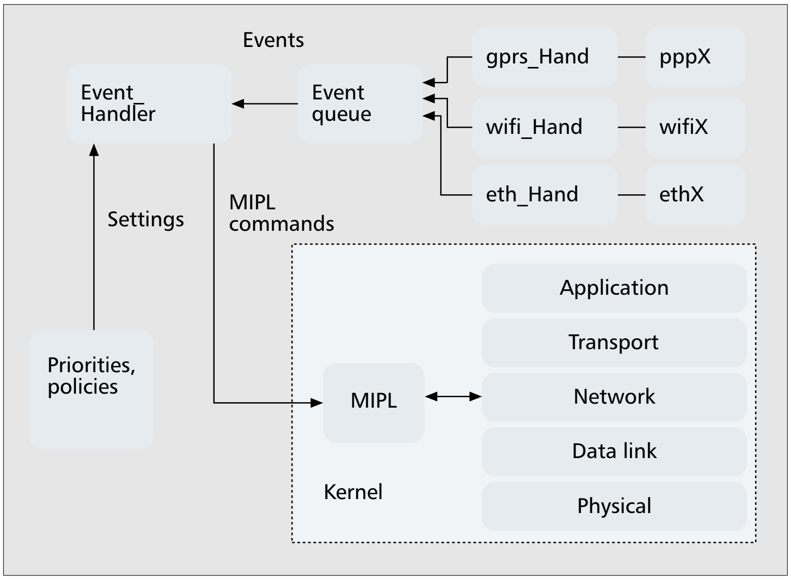


Figure 2. *Software architecture for lower-layer triggering.*

ponent of the total latency, especially in forced handoff. Cross-layer mobility management proposals reduce the detection delay using link layer information (e.g., signal strength). Besides the reduction in handoff latency, there are other reasons to employ lower-layer information:

- "Mobility policies" for mobile hosts can be implemented more easily when information about the status of the connection is reported promptly to the policy enforcing module.
- Lower layer monitoring may improve the behavior of higher layers of the network stack. Future network applications could be designed in order to take advantage of information concerning connectivity status.

The idea has been extensively tested in horizontal handoffs. Access points and a dedicated medium access control bridge are jointly used in [9] to reduce the handoff latency in WLANs. The micromobility management system S-MIP [10] extends Mobile IPv6 architecture by adding a new entity, the *decision engine*, in charge of performing handoff decisions for intradomain roaming. The DE uses the signal strength obtained from the link layer to classify movement patterns as *linear*, *stationary*, and *stochastic*. The aim of this classification is to choose the optimal schedule for the handoff and avoid ping-pong effects between the old and new access networks. Both approaches are limited to intradomain mobility in a homogeneous network.

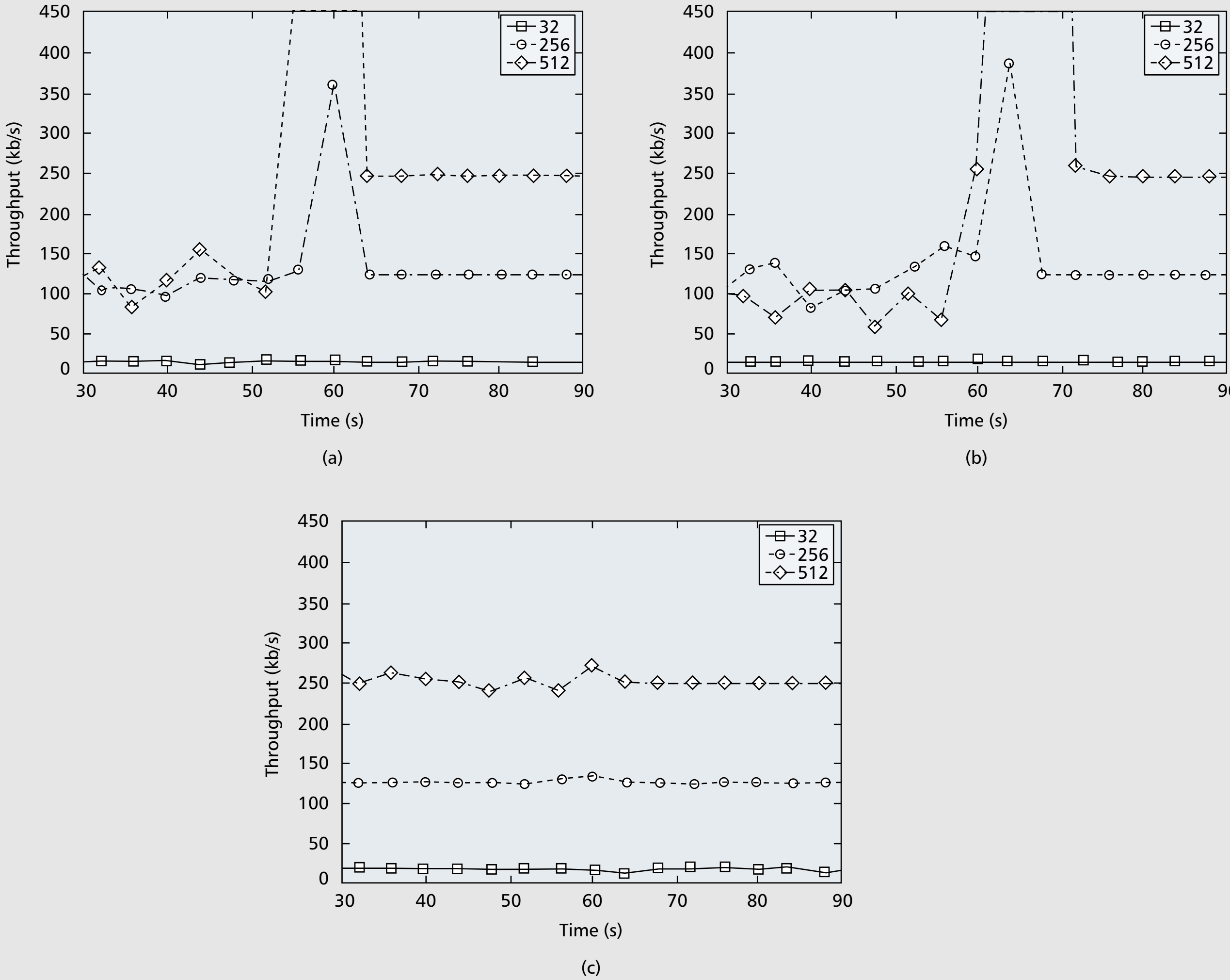
For heterogeneous networks there is the additional problem of defining a metric for the different interfaces. A model based on cost functions is presented in [11], where it is observed that further performance studies are required. The solution proposed in [12] uses link layer information to detect an interdomain handoff between heterogeneous networks, which are integrated through a network internetworking agent (NIA) that handles authentication, billing, and mobility management issues. A similar cross-layer approach, not Mobile IP-based, is present in the connection manager (CM)/virtual connectivity (VC) architecture [13]. The two modules are in charge of detecting network interface status and maintaining the continuity of ongoing connections during handoff, respectively, without Mobile IP support. The CM uses fast Fourier transform (FFT)-based decay detection schemes specifically adapted to the WLAN/cellular network roaming scenario. The VC module introduces a new protocol for mobility signaling, consisting of new TCP options, and an intermediate header between UDP and IP for UDP packets. An additional entity, the subscription/notification service is required in the network to support mobility in all possible cases. Even if the CM/VC approach has good experimental performance, it is demanding with respect to both changes in the end hosts (the network stack must be modified) and the network infrastructure (for the addition of the S/N service). This probably prevents a large-scale deployment of this solution.

A general cross-layer solution for mobility in heterogeneous networks requires the following components:

- Interface monitoring modules, specific for each network technology. For horizontal handoffs, the access routers can be responsible for link layer monitoring and handoff decisions, whereas for heterogeneous networks, the link layer monitoring must be performed at the MN.
- A decision module in charge of receiving link layer information and user-specified preferences, in order to trigger the handoff according to a decision algorithm.
- A handoff protocol (e.g., Mobile IPv6).

We present the results of an implementation of such an approach for Mobile IPv6 and vertical handoffs. Our prototype, represented in Fig. 2, runs in the Linux environment and currently supports Ethernet, IEEE 802.11b, and cellular network interfaces [14].

The *Event_Handler* runs in user space and is initialized by a description of the priorities of the network interfaces. It receives events from an *event queue*, where events are inserted by modules monitoring each different type of network interface. Events can regard either link presence/failure (e.g., the presence of an AP) or link quality. Potential parameters for link quality are specific to network technology and include signal strength, signal-to-interference ratio (SIR), bit error rate, and frame error rate. The decisions of the *Event_Handler* can take into account also power consumption, as MNs are frequently battery-powered, and different policies can be enforced by the *Event_Handler* command interface. Note that in our extensible architecture, the *Event_Handler* can use several different algorithms for handoff decisions, and switch among them at runtime. The results collected in our testbed were obtained using a simple two-threshold algorithm: the handoff toward the lower-priority interface (in our case, the cellular network) is triggered when the higher-priority signal strength (i.e., the WLAN) drops below the lower threshold. Triggering of the handoff in the opposite direction requires that the higher-



■ **Figure 3**. *TCP throughput for handoff from cellular network to WLAN 802.11b: a) GPRS; b) 100 kb/s UMTS; c) 384 kb/s UMTS.*

priority signal raises above the higher threshold. The double threshold algorithm is simple, but effective in inducing a hysteresis phenomenon that avoids a ping-pong effect.

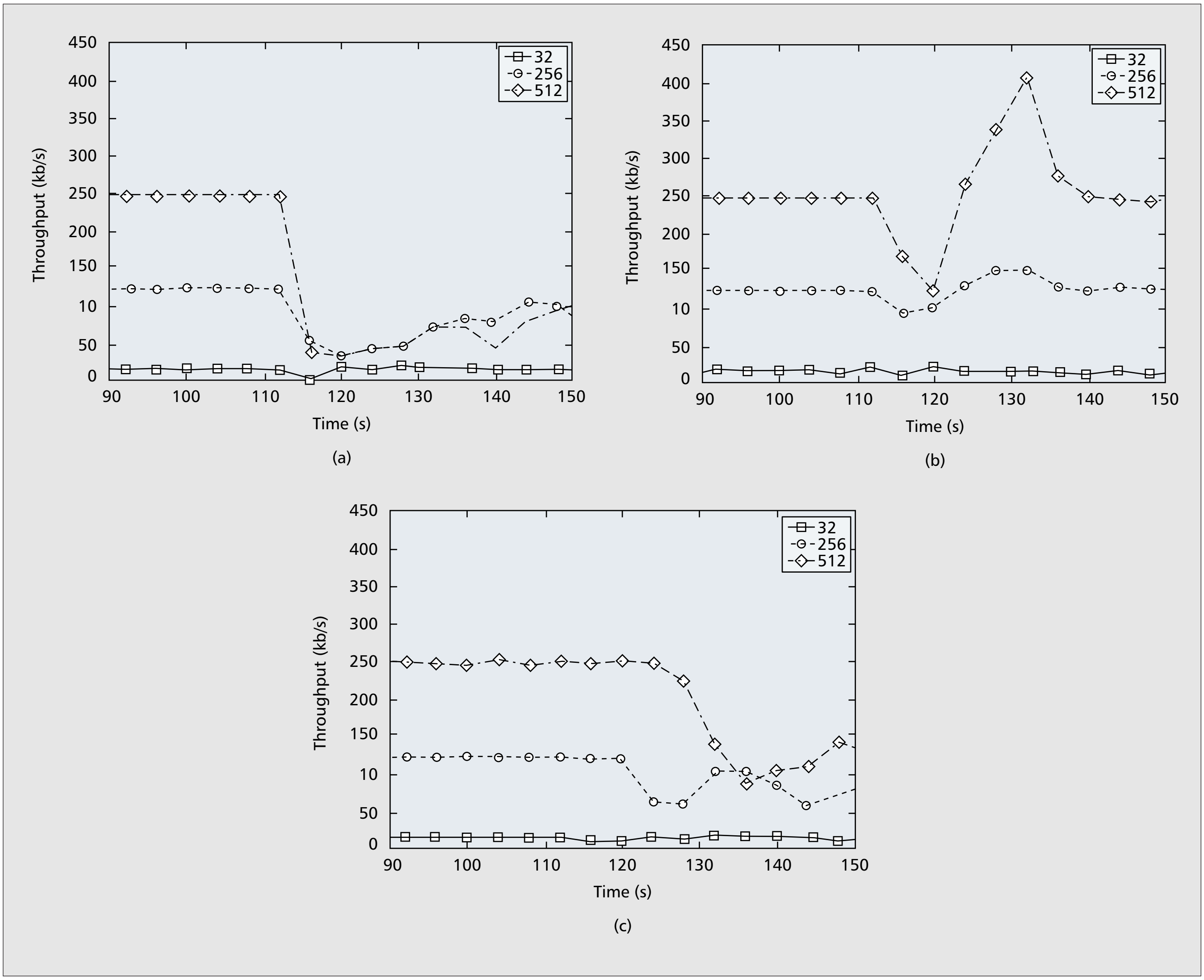
The results of link layer triggering usage in our testbed show considerable performance improvements for handoffs from higher-priority interfaces with packet loss. The overall handoff delay at the network layer is reduced to (about) 20 percent of the original time. In the next section we analyze the impact of this reduction of handoff delay at the transport level to argue that lower-layer triggering is crucial for mobility in heterogeneous networks.

End-to-End Performance Evaluation

In order to complete our discussion about seamless internetworking in mobile environments, we present some results about perceived end-to-end performance during handoff from the application point of view. We gathered extensive experimental data from our testbed, but for space reasons we focus here on TCP, and limit our attention to throughput as a performance index. Although it cannot be considered by any means a complete performance characterization, the study of TCP dynamics during handoff allowed us to draw some interesting and encouraging conclusions on the possibility of managing the handoff process seamlessly. Measurements were performed using the Distributed Internet Traffic Generator (D-ITG) (http://www.grid.unina.it/software/ITG), an innovative tool for network performance evaluation. The tool can be configured to generate constant bit rate traffic by tuning two parameters: packet size (PS) and interdeparture time (IDT). In our tests the D-ITG sender is the CN, whereas the D-ITG receiver is the MN. To take into account the declared bit rate of the cellular network technologies in use, we introduced the following *traffic load* classes:

- *Low*: PS = 32 bytes and IDT = 1/100 s, corresponding to a bit rate of 25.6 kb/s
- *Medium*: PS = 256 bytes and IDT = 1/100 s, corresponding to a bit rate of 204.8 kb/s
- *High*: PS = 512 bytes and IDT = 1/100 s, corresponding to a bit rate of 409.6 kb/s

Several trials were performed in the same operating conditions for each traffic class. The values reported in the plots represent an average value obtained on the receiver side (MN) across three repetitions of the test.



■ **Figure 4**. *TCP throughput for handoff from WLAN 802.11b to cellular network: a) GPRS; b) 385 kb/s; c) 100 kb/s UMTS with L2 triggering.*

Figure 3 shows the TCP throughput for the handoff from cellular networks to WLAN. The difference between the second and third cases is the speed of UMTS connection (100 kb/s and 384 kb/s, respectively). The handoff happens at time ≈60 s. In situations with higher traffic loads, the throughput increases when migrating to the WLAN. In particular, after handoff to the faster network, there is a transient high peak when the sending buffer is flushed. After that, a steady transmit rate is achieved. Note that a steady throughput is achieved with 384 kb/s UMTS for all the considered traffic loads.

In Fig. 4 the forced handoff from WLAN to cellular networks is shown. The handoff happens at time ≈ 120 s. The first plot is from WLAN to GPRS with L3 triggering where the handoff is triggered by forcing the wireless card to disassociate from the access point. The throughput undergoes a sharp reduction, also at low traffic loads, for about 20 s. The second plot refers to 384 kb/s UMTS (the handoff is performed in the same way). The throughput reduction lasts about 10 s, but throughput is always over 100 kb/s. At high traffic load there is a peak after the handoff, when TCP empties the sending buffer. The third plot refers to a 100 kb/s UMTS connection with an L2 triggered handoff. In this case handoffs were performed in a more realistic way by walking away from the access point until the L2 module in charge of monitoring the signal strength triggered the handoff. Note that in this case the perceived bit rate at the application level starts decreasing before the handoff due to the fading wireless signal. In general, L3 triggering becomes not very reliable when the wireless signal weakens. Packets carrying RAs may be lost, thus causing the handoff toward the cellular network, but the possible arrival of subsequent RAs from the WLAN determine a ping-pong effect between the two networks that make meaningless the measured throughput.

The handoff delay at the network layer is only a fraction of the overall handoff delay. Packets lost or arriving out of order during the handoff execution trigger TCP congestion avoidance mechanisms. Reliable protocols thus amplify the delay of the network layer. The mechanisms and size of this amplification are described below for the two most common cases.

User Handoff from Cellular Networks to WLANs

When the handoff is toward a faster link, packets sent on the WLAN arrive before packets sent earlier that are still traveling on the cellular network. The effect of this on the TCP protocol is twofold:

- At the sender end (the CN in our scenario), ACKs sent on the WLAN arrive very quickly, causing a higher transmission rate (see the peaks in Fig. 3).
- At the receiver end, packets sent on the WLAN arrive before previous packets sent on the cellular network.

TCP behavior depends on the implementation: New Reno TCP with SACKS, as implemented in Linux, acknowledges these packets selectively, causing resending of previous packets that are still traveling on the cellular link.

No packet is lost, since this is a soft handoff. However, a certain number of packets are retransmitted. For a traffic load of 25.6 kb/s, the handoff is smooth on both GPRS and UMTS links, and no packet is retransmitted. When the traffic load is 409.6 kb/s, the average number of retransmitted packets is 27 for GPRS and 4 for UMTS. Nevertheless, since packets are retransmitted on the fast WLAN link, the delay due to packet resending is always less than 0.2 s. A smooth handoff can thus be expected at the application level. Results from our experimentation confirm, at least for high traffic loads, the simulation results of [15], where an analysis of this kind of handoff is reported.

Forced Handoff from WLANs to Cellular Networks

This handoff causes packet loss from the sender to the MN. Since TCP is a reliable protocol, it associates to every packet a retransmission timeout (RTO) for the acknowledgment based on the estimated RTT. When the timeout expires, the sender attempts retransmissions with an exponential backoff delay; after receipt of an acknowledgment, the sender assumes that the network is congested and enters a slow-start phase.

The negative side effect of this mechanism on handoff is that even when the handoff is completed at the network level, the retransmission of lost packets is not resumed until the backoff interval expires. The average value of this additional TCP delay grows exponentially with the number of retransmission attempts, which depends on the handoff delay at the network level. TCP delay is thus exponentially dependent on network delay. With L2 triggering it is possible to substantially reduce the drawback of the exponential backoff, as the delay between the last ACK on the WLAN and the first ACK on the UMTS network becomes about 300 ms, mostly due to the increased RTT. However, even if with L2 triggering no packet is actually lost at the network layer, the last packet sent on the WLAN is considered lost by the CN, since the RTO computed on the fast WLAN link is much smaller than the new RTT. The packet is then retransmitted on the UMTS link. Thus, even with L2 triggering, the TCP of the CN goes into slow-start mode. This is an inherent problem of TCP when passing to a slower link, but since the capacity of the new link is still unknown, the slow-start procedure is probably the best way to achieve optimal throughput. In Fig. 5 delay composition for a traffic load of 409.6 kb/s is shown.

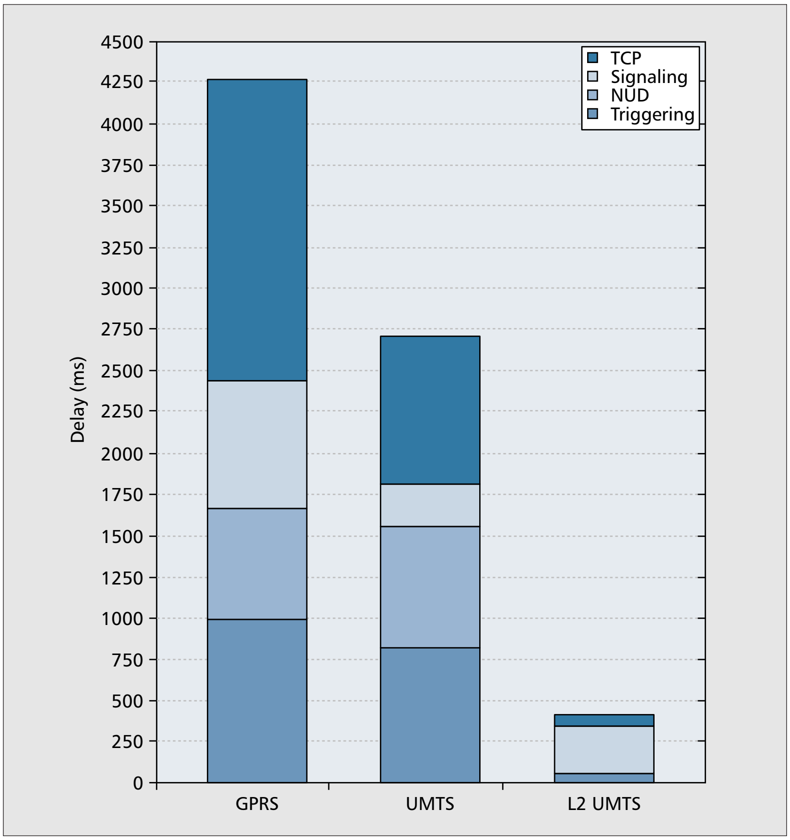


Figure 5. *Delay composition.*

Conclusion

Hybrid mobile networks like those formed by WLANs and 3G cellular data networks require efficient handoff mechanisms to guarantee seamless connectivity. In this work we tested a loosely coupled approach to internetworking between WLAN and cellular networks. The results show that the higher bandwidth of present UMTS networks allows traffic loads up to 400 kb/s to be transferred to WLANs without altering the bit rate. This handoff is really seamless at the application level. Even for GPRS connections, handoff toward WLAN can be performed without packet loss. The only consequence at the application level is a change in the perceived bit rate. We show how the employment of L2 triggering allows seamless handoffs at the network layer also when passing from WLAN to cellular networks. In this case, however, due to the different RTT on the two links, the TCP incurs in the slow-start phase, causing a temporary drop in the bit rate at the application level. Since L2 triggering avoids the TCP exponential backoff, the disruption of the connection can be reduced in this kind of handoff from a few seconds to about 0.4 s. This is an encouraging result for the loosely coupled approach and Mobile IPv6, particularly because it is obtained with no change of the network infrastructure.

Since the L2 triggering avoids the TCP exponential back-off, the disruption of the connection can be reduced, in this kind of handoff, from a few seconds to about 0.4 s. This is an encouraging result for the loosely coupled approach and the Mobile IPv6 protocol.

Acknowledgments

This work was funded in part by the Minstero dell'Istruzione, dell'Università e della Ricerca (MIUR) with the FIRB project WEB-MINDS and by the European Community's Sixth Framework Programme with the IST FP6 Network of Excellence, E-NEXT.

References

[1] S. McCann and H. Flygare, "Hiperlan/2 Public Access Interworking with 3G Cellular Systems," *Wireless Networks*, vol. 10, no. 1, 2004, pp. 43–51.

[2] M. Buddhikot *et al.*, "Integration of 802.11 and Third-Generation Wireless Data Networks," *Proc. IEEE INFOCOM '03*.

[3] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.

[4] N. Montavont and T. Noel, "Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN," *Mobile Networks and Apps.*, vol. 8, no. 6, 2003, pp. 643–53.

[5] R. Hsieh and A. Seneviratne, "A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP," *Proc. ACM MobiCom '03*, pp. 29–41.

[6] R. Chakravorty *et al.*, "Practical Experience with Wireless Networks Integration Using Mobile IPv6," *MobiCom '03 Poster, ACM Mobile Comp. and Commun. Rev.*, vol. 7, no. 4.

[7] X. Zhao, C. Castelluccia, and M. Baker, "Flexible Network Support for Mobility," *Proc. ACM MobiCom '98*, pp. 145–56.

[8] H. Luo *et al.*, "Integrating Wireless LAN and Cellular Data for the Enterprise," *IEEE Internet Comp.*, vol. 7, no. 2, 2003, pp. 25–33.

[9] H. Yokota et al., "Link Layer Assisted IP Fast Handoff Method over Wireless LAN Networks," *Proc. ACM MOBICOM '02*, pp. 131–39.

[10] R. Hsieh *et al.*, "S-MIP: A Seamless Handoff Architecture for Mobile IP," *Proc. IEEE INFOCOM '03*, vol. 3, 2003, pp. 1774–84.

[11] J. McNair and F. Zhu, "Vertical Handoffs in Fourth-Generation Multinetwork Environments," *IEEE Wireless Commun.*, June 2004, pp. 8–15.

[12] I. F. Akyildiz, J. Xie, and S. Mohanty, "A Survey of Mobility Management in Next-Generation All-IP-Based Wireless Systems" *IEEE Wireless Commun.*, Aug. 2004, pp. 16–23.

[13] Q. Zhang *et al.*, "A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks," *IEEE JSAC*, vol. 22, no. 5, 2004.

[14] M. Bernaschi, F. Cacace, and G. Iannello, "Vertical Handoff Performance in Heterogeneous Networks," *Int'l. Wksp. Mobile and Wireless Net.*, Montreal, Canada, Aug. 2004.

[15] W. Hansmann, M. Frank, and M. Wolf, "Performance Analysis of TCP Handover in a Wireless/Mobile Multi-Radio Environment," *Proc. IEEE LCN '02*.

Biographies

MASSIMO BERNASCHI graduated in physics in 1987 from Tor Vergata University, Rome, Italy. After that he joined the IBM European Center for Scientific and Engineering Computing (ECSEC), Rome. He spent 10 years with IBM working in the field of parallel and distributed computing. Currently he is with the Italian National Research Council (CNR) as head of the technology office in the Institute for Computing Applications. Moreover, he is an adjunct professor of computer science at La Sapienza University in Rome.

FILIPPO CACACE (f.cacace@unicampus.it) graduated in electronic engineering from Politecnico di Milano in 1988 and received a Ph.D. in computer science. His research interests include wireless and heterogeneous computer networks, network applications, database programming languages, and logic programming. He is currently working at the University Campus Bio-Medico, Rome, Italy.

GIULIO IANNELLO graduated in electronic engineering from Politecnico di Milano in 1981 and received a Ph.D. in computer science from the University of Napoli Federico II in 1987. Currently he is a full professor of computer science and computer engineering at the University Campus Bio-Medico in Rome. His primary research interests include wireless and high-performance computer networks, multimedia data processing, design and analysis of parallel algorithms, and performance evaluation of parallel and distributed systems.

ANTONIO PESCAPE [M] received his M.S. Laurea degree and Ph.D., both in computer engineering and systems, from the University of Napoli Federico II. His research interests are in the networking field with focus on models and infrastructure for QoS over IP networks, models and algorithms for Internet traffic, and network measurement and management.

STEFANO ZA graduated in computer science in 2004 from La Sapienza University, Rome. Currently he is with the wireless network research sector of University Campus Bio-Medico. His research interests include wireless and high-performance computer networks.