

Providing UMTS connectivity to PlanetLab nodes

Alessio Botta, Roberto Canonico, Giovanni Di Stasi, Antonio Pescapè, Giorgio Ventre
University of Napoli Federico II, Via Claudio 21, 80125, Napoli, Italy
Consorzio Interuniversitario Nazionale per l'Informatica CINI
Laboratorio ITeM-CINI, Via Cinthia 4, 80126, Napoli, Italy
{a.botta,roberto.canonico,giovanni.distasi,pescapè,giorgio}@unina.it

ABSTRACT

Planetlab is widely recognized as being one of the most important Internet-scale testbeds. However, while allowing experimentations involving hundreds of hosts spread all over the world, PlanetLab still suffers of a few significant limitations. One of these limitations is the lack of heterogeneity, in particular in terms of access technologies. In this paper we describe the efforts we made, in the context of the OneLab European project, in order to mitigate this problem. In particular, we describe how we managed to integrate UMTS connectivity into a PlanetLab-based testbed. We illustrate the technical challenges we had to face, the final result we obtained, and present a case study meant to show the utility of having UMTS connectivity available in Planetlab.

1. INTRODUCTION

PlanetLab is a geographically distributed testbed for deploying and evaluating planetary-scale network applications in a highly realistic context. Since its initial deployment in 2003, it has become of indisputable importance for the academic research community and industry, which use such infrastructure for the evaluation of a wide range of distributed systems. Nowadays the testbed is composed of more than 900 nodes, hosted by about 470 academic institutions and industrial research labs.

One of the main limitations of such infrastructure is the lack of heterogeneity, that jeopardizes the effectiveness of PlanetLab in providing realistic results from the experimentations [1], [2]. Nearly all PlanetLab nodes are server-class computers connected to the Internet through high-speed research or corporate networks. This contrasts with the real Internet in which

the connected devices also comprise laptops, hand-held computers, cellular phones, network attached storage devices, video-surveillance appliances, as well as other household devices. Besides, all such systems are connected to the Internet through a variety of access technologies, both wired (ADSL, CATV) and wireless (WiFi, GPRS, UMTS, CDMA, etc.). To introduce more heterogeneity into the testbed, several approaches have been pursued. Dischinger et al., for instance, have recently proposed an architecture (SatelliteLab) to include into the experiments a variety of non-dedicated devices made available by residential users [3].

OneLab is a European Project, funded by the European Commission in its Sixth Framework Programme [4]. The project started in September 2006 with “*two overarching objectives: (1) to extend the current PlanetLab infrastructure and (2) to create an autonomous PlanetLab Europe*”. PlanetLab Europe is a European-wide research testbed that is linked to the global PlanetLab through a peer-to-peer federation [6].

The objective of extending PlanetLab has been pursued in OneLab through development of a number of extensions to the PlanetLab architecture, aimed at enriching it with new wireless access technologies, with powerful traffic monitoring instruments and with new emulation capabilities. These new features have been implemented and validated in a project-wide distributed testbed, the so-called *Private OneLab*, whose resources are provided by and accessible to OneLab members only. Private OneLab serves as an incubator of new features to be eventually included in the public infrastructure. As these new features are thoroughly tested and validated, they can be later moved into the public PlanetLab Europe testbed.

To add more diversity to PlanetLab, the OneLab project integrated different kinds of access technologies (such as WiFi, UMTS, WiMax) into Private OneLab. In this paper we describe how we managed to integrate UMTS connectivity into *Private OneLab*. In particular, we describe the design, implementation, and testing of our solution, describing in details the technical challenges we had to face and the way we solved them. We

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ROADS'08, December 9, 2008, Madrid, SPAIN

Copyright 2008 ACM 978-1-60558-266-5/08/0012 ...\$5.00.

believe this can be useful for other researchers working on PlanetLab as well as on other distributed testbeds. Moreover, we present a proof of concept experiment aimed at characterizing the UMTS wireless link. We provide results related to this experiment and some insights we gained from their analysis. Since our technique is quite general and perfectly integrated into the core of the PlanetLab architecture, we believe that our approach can be also pursued to extend PlanetLab with support for other networking technologies.

2. UMTS INTO PLANETLAB

In order to introduce more heterogeneity into the PlanetLab architecture we integrated UMTS connectivity. More precisely, our aim was to provide every node of the testbed with the possibility of using a UMTS interface and to handle the related Point-to-Point Protocol (PPP) connection. While this task can be easy in a common Linux box, it is not as easy in a PlanetLab node. This is because such nodes have a modified kernel and also because very limited privileges are granted to normal users. In this section we will firstly explain why we decided to integrate UMTS connectivity into PlanetLab and then how we did it.

2.1 Motivations

UMTS, Universal Mobile Telecommunications System, is a third-generation (3G) cellular phone technology standardized by 3GPP [8]. This standard has been adopted by several telecom operators all over the world, and it has become largely popular to provide both voice services and ubiquitous high-speed access to the Internet. The number of UMTS users reached 400 millions and it is still increasing [10]. Most recent releases of the UMTS standard support both voice traffic and data traffic in a common IP-based network infrastructure. In such releases, also thanks to the introduction of the High Speed Downlink/Uplink Packet Access (HSDPA and HSUPA), data rates can be as high as 14 Mbps downstream and 5.8 Mbps upstream [9].

The choice of integrating UMTS connectivity in our PlanetLab-based testbed was motivated by different reasons. Firstly, UMTS has become a competitor of other well known access technologies such as WiFi and xDSL. A second reason is related to the applications that are spreading over such network infrastructure. The IP Multimedia Subsystem (IMS) [11], which is being progressively introduced in the UMTS architecture, is triggering the development of new generations of network applications such as presence, conferencing and location-based services. Such applications are attracting an increasing number of users and should be therefore taken into account in a realistic network testbed.

It is also worth noting that our main goal was not to integrate a specific UMTS network into PlanetLab,

but rather to allow PlanetLab institutions to equip their nodes with such kind of connectivity using a Telecom Operator of choice. In principle, this allows to perform experiments by using the UMTS connection provided by different networks and to compare the results. In fact, to accomplish our task in OneLab we have utilized two UMTS networks: a private UMTS micro-cell provided by Alcatel-Lucent Italia and located at their 3G Reality Center in Vimercate (Italy), and a commercial connection provided by one of the principal European telecom operators. While the first network is private and could be used only thanks to the collaboration of Alcatel-Lucent Italia, the second is a public UMTS network, that we used without any particular agreement with the operator.

2.2 Usage model

In order to run their experiments, PlanetLab users have to acquire a *slice*, that is a collection of virtual machines instantiated into a subset of the PlanetLab physical nodes. Slices run concurrently on PlanetLab, and act as network-wide containers providing isolation among the experiments. The virtualization technology currently adopted by PlanetLab is Linux VServer¹. VServer implements a soft partitioning concept based on Security Contexts, which allows to create many independent Virtual Private Servers on a single physical node.

Before even approaching the problem of integrating UMTS support into the PlanetLab software system, we had to define a proper usage model for this specific extension, i.e. a proper model for sharing such a resource (the UMTS connection) in the PlanetLab virtualized environment [1]. To this purpose, we decided to adopt a policy that allows only one experiment (i.e. slice) at a time to control and use the UMTS interface. This is motivated by two main reasons: (i) the low bandwidth of a UMTS connection would cause high interference between concurrent experiments, and (ii) in order for the experiment to be realistic, the dial-up connection should be set up and torn down just before and after the test respectively.

To foster the worldwide deployability of UMTS in PlanetLab, we chose to support two of the most widely used network interface cards (at the beginning of our work): the Option Globetrotter GT+ 3G card² and the Huawei e620 card³.

The nodes equipped with a UMTS interface are also provided with an Ethernet NIC. The wireless UMTS connection is used for the experiments while the wired Ethernet is used for control data. This is also necessary because the UMTS connectivity provided by the

¹<http://linux-vserver.org>

²<http://www.option.com>

³<http://www.huawei.com>

operators often employ firewalls or filters that do not allow to reach the UMTS-equipped host by using terminal services such as *ssh*. When the UMTS connection is established, the default route is left to the Ethernet interface. In order for a slice to use the UMTS connection it is necessary to specify the destinations for which the UMTS connection is required or to explicitly bind to the UMTS interface. The control of the UMTS connection is provided to the users by means of a special *umts* command, which allows users to start, stop, and check the status of the connection, as well as to add and delete the destinations previously mentioned. These commands should perform some operations requiring superuser (i.e. root) privileges, which is not allowed from inside a slice. To overcome this problem we use the features provided by *vsys* [13], as explained in the following section.

2.3 Implementation issues

The PlanetLab node operating system is derived from a Fedora Core Linux distribution (Fedora Core 8 as of August 2008) with some patches. The patches are mainly targeted to the virtualization features needed to support the concurrency of the experiments and the related networking issues. To add support for the UMTS interfaces we needed to add both kernel modules and user-space tools.

The kernel modules we had to integrate into the distribution are those related to the management of the PPP connection (*ppp-generic*, *ppp-filter*, *ppp-async*, *ppp-sync-tty*, *ppp-deflate*, and *ppp-bsdcomp*) and those required by the two NICs, i.e. *pl2303* and *usbserial* for the Huawei card, and *nozomi* [18] for the Globetrotter card. The *nozomi* module required some modifications in order to run with the latest PlanetLab kernel, based on Linux kernel version 2.6.22.

In user space some tools are needed to establish the PPP connection and then to discipline the access to it. The first goal is achieved thanks to the *comgt* [15] and *wvdial* [16] tools. *comgt* is used to register into the operator network. *wvdial* establishes the actual PPP connection. The second goal is achieved, instead, by means of the *iproute2* and *iptables*. Their use is aimed at isolating the slice that requested the UMTS interface (*UMTS slice* in the following), by allowing only that slice to use the UMTS interface, without interfering with ongoing experiments in other slices. In particular, *iproute2* is used for (1) creating an additional routing table, which contains only a default route that points to the UMTS interface, and for (2) defining the rules that allow only specific packets to be routed through the UMTS interface using the additional table. Such packets are those matching one of the following rules: i) those generated by the *UMTS slice* and directed towards the destinations for which the user requested

UMTS interface; ii) those generated by the *UMTS slice* and having source address equal to that of the UMTS interface. Packets generated by the *UMTS slice* are recognized by means of a mark applied with *iptables*, exploiting a feature of the new VNET+ subsystem of Planetlab [12]. Packets generated by other slices will not be able to use the UMTS interface (as they are either not marked or marked differently) and will continue to get routed following the rules contained in the default routing table. In order to enforce the isolation between slices (i.e. allowing only one slice at a time to use the UMTS connection), we also add another rule with *iptables* that drops all the packets not generated by the *UMTS slice* and that are about to go out using the UMTS interface. Such rule is necessary to cope with special cases, such as a user of a different slice that tries to generate packets destined to the other endpoint of the PPP connection or that binds an application to the UMTS interface.

All these user space tools require superuser privileges and, therefore, must run in the root context of a PlanetLab node. For PlanetLab users, however, it is not possible to execute commands with such privileges. To overcome this problem, in PlanetLab there is a utility called *vsys* [13] which allows to execute a program with root privileges from inside a slice. Basically, *vsys* creates a pair of FIFO pipes that allow a program in a slice (i.e. the front-end program) to communicate with another program in the root context (i.e. the back-end program). The front-end program receives the commands requested by the user, and pass them to the back-end program, that runs in the root context and executes the aforementioned user space tools. The front-end program implements the following features:

- *start*: check and lock the UMTS interface, set up the UMTS connection, and enforce the routing rules;
- *stop*: tear down the UMTS connection, unlock the interface, and delete the routing rules;
- *status*: check the status of the connection;
- *add destination*: add a rule for this destination to be reached through the UMTS connection;
- *del destination*: delete the rule associated to this destination.

All the software we used and developed is publicly available in the OneLab software repository [7]. As soon as it will be moved into the PlanetLab Europe (PLE) software repository, it will be possible to add UMTS connectivity at each PLE site.

3. A PROOF OF CONCEPT EXPERIMENT: ANALYSIS OF A COMMERCIAL UMTS CONNECTION

In these experiments we show how it is possible to perform a simple analysis of a UMTS link using PlanetLab nodes and the tools we developed. In particular, we equipped one of our PlanetLab nodes with a commercial UMTS connection provided by one of the major operator in Italy. We then utilized the tools presented in the previous section, and we also generated synthetic traffic towards another PlanetLab node. This allowed us to perform an initial characterization of a UMTS connection. To this aim we performed two kinds of experiments: (i) measuring the characteristics of the end-to-end path between a mobile node with UMTS connectivity and fixed nodes located remotely at another site of PlanetLab (we will refer to this experiment as UMTS-to-Ethernet), and (ii) measuring the characteristics of the Ethernet path between the same two nodes (we will refer to this experiment as Ethernet-to-Ethernet). This is to compare the two links in order to assess their common and different properties.

3.1 Methodology

For the experiments, we generated probing traffic, between two nodes: a UMTS-equipped node located at our Laboratories in Napoli and an Ethernet-connected node located at INRIA in France. Traffic was generated by using D-ITG (Distributed Internet Traffic Generator) [19], a platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes for both IDT (Inter Departure Time) and PS (Packet Size) random variables (exponential, uniform, cauchy, normal, pareto, ...). D-ITG supports both IPv4 and IPv6 traffic generation and it is capable to generate traffic at network, transport, and application layer. Moreover, it can perform both one-way-delay (OWD), round-trip-time (RTT), packet loss, jitter, and throughput measurement. Finally, it is capable to store information about sent and received packets on both the sender and the receiver sides.

We compared the characteristics of the two end-to-end paths by using two traffic classes. The first one was obtained by generating a single VoIP-like flows made of 72 Kbps of UDP Constant Bitrate (CBR) traffic resembling the characteristics of a real VoIP call using codec G.711. The other one was obtained by using a 1-Mbps UDP CBR flow with packet size equal to 1024 Bytes and packet rate equal to 122 pps. Both flows lasted for 120 seconds. The former kind of traffic allows assess the feasibility of a voice call over the UMTS and to compare the achievable quality with that on a high capacity link. The latter kind, instead, allows to analyze the behavior of the UMTS up-link in fully saturated conditions.

After the traffic generations ended, we retrieved the log files from the two nodes and we analyzed them by means of ITGDec, another component of D-ITG suite.

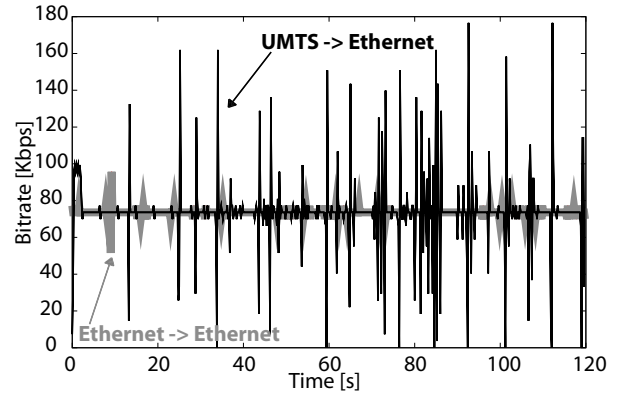


Figure 1: Bitrate of the VoIP-like flow.

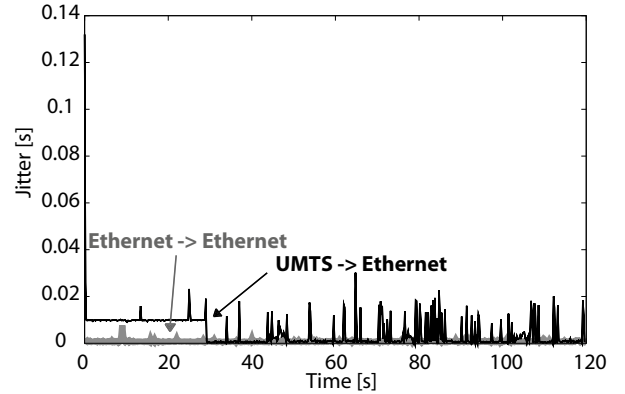


Figure 2: Jitter of the VoIP-like flow.

In this way we obtained the samples of four QoS parameters that are bitrate, jitter, loss, and round-trip time (RTT). Such samples represent the average values calculated over non-overlapping windows of 200 milliseconds. In the following subsections we show the time plot of the samples of such parameters obtained with the two traffic classes using the two network paths. It is worth underlining that each measurement experiment was executed 20 times and very similar results were obtained.

3.2 Results

3.2.1 VoIP traffic

Figures 1-3 show the time plots of the bitrate, jitter, and round trip time respectively. The packet loss is not reported for this experiment because it was always equal to 0. In each figure the results obtained with both the UMTS-to-Ethernet and the Ethernet-to-Ethernet paths are present. As we can see, all the parameters present different behaviors in the two cases.

In more details we observe that the bitrate of the UMTS connection is more fluctuating than in the Ethernet case even though, in both cases, the required value

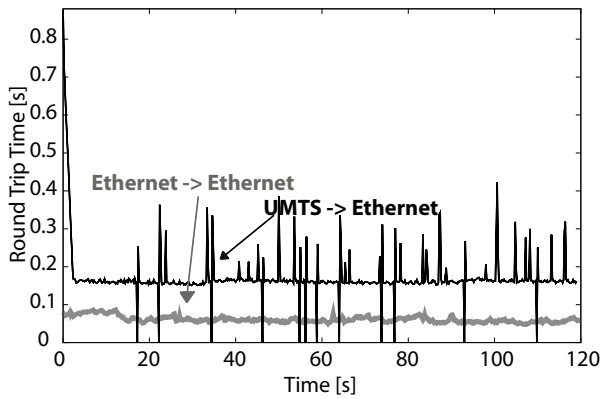


Figure 3: RTT of the VoIP-like flow.

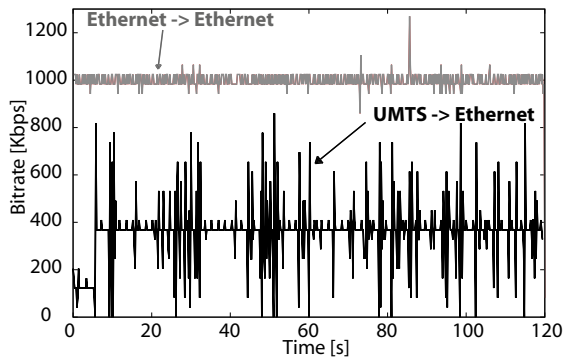


Figure 4: Bitrate of the 1-Mbps flow.

is achieved in average. The jitter plots, instead, show the UMTS connection introduces a higher jitter, which is also more fluctuating. It reaches values up to 30 milliseconds which, however, still allows a VoIP communication to be satisfying for the users. Finally, looking at the round-trip delay, we can observe that the average value is higher for the UMTS connection with respect to the Ethernet one. Moreover, as seen for the jitter, the RTT is more fluctuating on the wireless connection and it reaches values up to 700 milliseconds.

3.2.2 1-Mbps CBR traffic

Figures 4-7 show the bitrate, jitter, loss, and round trip delay we obtained on both end-to-end paths. This is to show the differences between the characteristics of these two network connections in the case of a 1Mbps flow, which clearly saturates the up-link of the UMTS connection but not the Ethernet. As a first consideration, we observe that the bitrate of the UMTS reaches a maximum value of around 400 Kbps. Such value is lower than the required one, and it is therefore representative of the maximum capacity of the up-link of the UMTS connection. The jitter, packet loss, and round-trip delay plots show the very low performance achieved by the UMTS connection. Such results are justified by

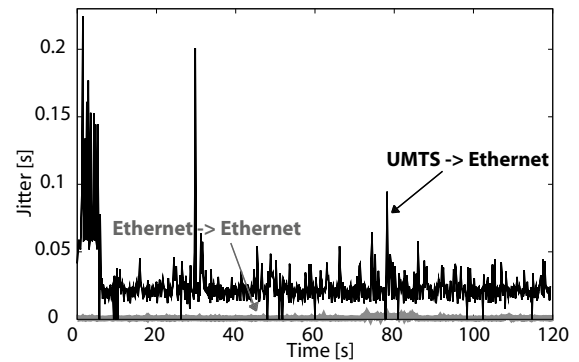


Figure 5: Jitter of the 1-Mbps flow.

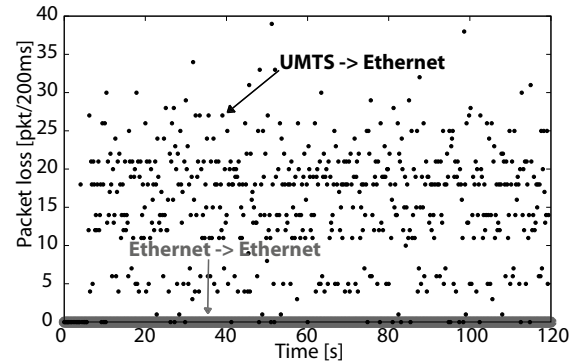


Figure 6: Loss of the 1-Mbps flow.

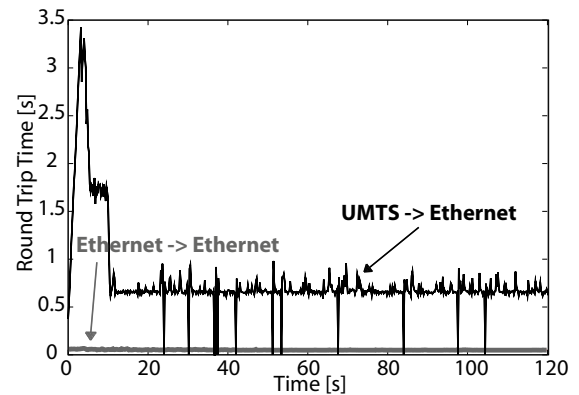


Figure 7: RTT of the 1-Mbps flow.

the fact that such connection is operating in very congested conditions in this case, and therefore all the QoS parameters are heavily affected. In particular we can see that the jitter reaches values larger than 200 milliseconds, which makes a real time communication nearly impossible. This is even more confirmed by the values of the RTT which can be as large as 3 seconds.

Besides, looking more closely at the figures we observe that for the UMTS connection there is a first time period in which the situation is much worse than in the rest of the time. In particular, if we look at Figure 4 we

can see that in the first 5 seconds the achieved bitrate is about 150 Kbps. After that time, instead, the bitrate is more than doubled. This is due to some sort of adaptation algorithm happening inside the UMTS network, which allocates the network resources to the users in a *on-demand* fashion. The change of link characteristics has also an effect on the other parameters, which have an increment of the performance after the first 5 seconds.

4. RELATED WORK

As far as the connectivity of PlanetLab nodes is concerned, the lack of heterogeneity has also been noticed by others [17]. The authors of this paper analyzed the inter-domain connectivity of PlanetLab nodes, noting that a large part of the hosts are connected through what they call the Global Research and Educational Network, a set of interconnected high-speed research networks. They also state that, for this reason, most of the end-to-end measurements between PlanetLab nodes are not representative of the real Internet characteristics. As a consequence, it has also been noted that the behavior of some applications on PlanetLab can be considerably different from that on the Internet [18]. A possible solution to this lack in PlanetLab has been recently proposed in [3]. In such paper the authors propose an enhancement to the PlanetLab architecture that allows to route the user traffic through external boxes with different access networks. However, such boxes are not part of the testbed but rather controlled by special PlanetLab nodes. Our approach differs from this one, since our objective was to introduce heterogeneity into the testbed itself, hence our main effort consisted in integrating UMTS support into the ordinary PlanetLab code.

5. CONCLUSIONS

PlanetLab has become very important for the research community and the industry in order to perform Internet-scale experimentations. However, one of its main problems is the lack of heterogeneity, especially in the access networks of the nodes, which translates into non highly realistic scenarios for the experiments. In this paper we presented the efforts we made to solve this problem integrating UMTS connectivity into PlanetLab testbed. We detailed the problems we encountered and the solutions we found. We also presented a case study illustrating a possible application of the features introduced. We believe this work will be useful for both developer and users of the PlanetLab testbed in order to be aware of the available features and exploit them as much as possible.

6. ACKNOWLEDGMENTS

This work has been funded by the FP6 IST-2005-034819 project “OneLab” and partially supported by the CONTENT EU Network of Excellence (IST-FP6-038423). This work was also partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the NADIR Project (PRIN Program).

7. REFERENCES

- [1] R. Canonico, G. Ventre, S. D’Antonio, “Extending the Planetlab Usage Model for Distributed Heterogeneous Research Testbeds”, ROADS07, Warsaw, Poland, July 2007. Paper available at <http://roads.cs.princeton.edu/warsaw07/abstract12.pdf>.
- [2] Himabindu Pucha and Y. Charlie Hu and Z. Morley Mao, “On the impact of research network based testbeds on wide-area experiments”. Procs. of IMC ’06, the 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeiro, Brazil, 2006.
- [3] M. Dischinger, A. Haeberlen, I. Beschastnikh, K. P. Gummadi, S. Saroiu, “SatelliteLab: Adding Heterogeneity to Planetary-Scale Network Testbeds”, ACM SIGCOMM 2008 conference, Seattle, WA, USA.
- [4] <http://www.onelab.eu/>
- [5] <http://www.planet-lab.eu/about>
- [6] <http://planet-lab.org/federation>
- [7] <http://build.one-lab.org/>
- [8] <http://www.3gpp.org/>
- [9] http://www.ericsson.com/technology/tech_articles/HSPDA.shtml
- [10] <http://www.umts-forum.org/content/view/2463/174/>
- [11] G. Camarillo, M.-A. Garcia-Martin, “The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds”, John Wiley & Sons, 2006.
- [12] <http://www.cs.princeton.edu/~sapanb/vnet/>
- [13] <http://www.cs.princeton.edu/~ensuremath{\sim}sapanb/vsys/>
- [14] <http://www.pharscape.org/>
- [15] <http://sourceforge.net/projects/comgt>
- [16] http://freshmeat.net/redirect/wvdial/11770/url_homepage/index.php
- [17] S. Banerjee, T. G. Griffin, M. Pias. “The Interdomain Connectivity of PlanetLab Nodes”, Passive and Active Measurement Conference, Antibes Juan-les-Pins, France, Apr 2004.
- [18] J. Ledlie, P. Gardner, M. Seltzer, “Network Coordinates in the Wild”, NSDI07, Cambridge, MA, Apr 2007.
- [19] <http://www.grid.unina.it/software/ITG/>