QoS enabled mobility support for mesh networks

(Invited Paper)

Dario Gallucci*† and Silvia Giordano†
*Dipartimento di Automatica ed Informatica, Politecnico di Torino, Italy
Email: dario.gallucci@polito.it
†University of Applied Sciences (SUPSI), Galleria 2, CH-6928 Manno, Switzerland
Email: [dario.gallucci, silvia.giordano]@supsi.ch

***Abstract*—Existing solutions for building wireless mesh networks suffer from reduced efficiency. This is due to lack of reliable self-configuration procedures that can dynamically adapt to varying network conditions, lack of efficient and scalable end-to-end Quality of Service (QoS) support, and lack of generalized and seamless mobility support. Therefore, ensuring a certain level of QoS to mobile users is one of the most crucial challenges, and it is clearly one of the major inhibitors to the commercial success of mesh networks. In this paper, we present a heuristic service classification approach and a correspondent traffic classification algorithm for handling the multimedia traffic at the network and the application layer, in order to better characterize the traffic. The preliminary simulation results show that the approach is able to classify a large part of the traffic and, consequently, to handle it appropriately. This is confirmed by the encouraging preliminary results where the worst case is managed fairly in comparison to the average case.**

I. Introduction

The last few years have seen an exceptional growth of the Wireless Local Area Network (WLAN) industry, with a substantial increase in the number of wireless users and applications. This growth was due, in large part, to the availability of inexpensive and highly interoperable network solutions based on the Wi-Fi standard [1], and to the growing trend of providing built-in wireless network cards into mobile computing platforms. Today, public and private organizations are developing wireless mesh networks, peer-to-peer multi-hop wireless networks based on wi-fi interconnection, in which participant nodes connect with redundant interconnections and cooperate with one another to route packets. However, there are several business and technological challenges that have to be addressed to make those Wi-Fi based networks a global network infrastructure [2]. In particular, as the number of mobile Internet users increases and new emerging applications appear, ensuring a level of Quality of Service (QoS) to them that is not too far from the one experienced by wired Internet users in terms of application reliability, throughput, end-to-end delay bounds, etc., is one of the most crucial challenges, and it is clearly one of the major inhibitors to a real success of those solutions. In particular, existing solutions for building wireless mesh networks suffer from reduced efficiency due to lack of reliable self-configuration procedures that can dynamically adapt to varying network conditions, the lack of efficient and scalable end-to-end QoS support, and the lack of generalized and seamless mobility support. Further, as opposed to the case of the wired Internet, the convergence between data and multimedia networks is not really happening. Thus now, while several Internet users are benefiting from tools such as Skype for low cost voice services, or such as Youtube for low cost video services, there is no counterpart for mobile networks. The reasons for this can be found in:

- the lack of ubiquitous coverage for mobile users. A mobile user can work from a node of mesh network, but cannot easily move, as there is no automatic handover at the network layer.
- Even if some dedicated solution for automatic handover is provided, there is no assurance that the applications will continue to work and/or will not experience any difficulty.
- Furthermore, when seamless handover can be provided, it can be hardly provided for heterogeneous networks (e.g. benefiting of GPRS or UMTS wide coverage where the mesh network is not available).

Global node mobility requires supporting seamless vertical (among different type of networks) and horizontal (among networks of the same type) handovers, either between different domains (inter-domain handoff or macro-mobility), or inside the same domain (intra-domain handoff or micro-mobility). Recent work in this area includes layer 3 approaches supporting intra-domain handoff in wireless mesh networks and mechanisms for intra and inter-domain handoff. They can be potentially operated between different networks, using multicast groups to coordinate transfer connection decisions, and client-side network layer mobility modules to support both micro and macro-mobility. Support for applications with QoS constraints cannot be achieved with the handoff schemes adopted by these solutions, which are based on simple threshold approaches that initiate a handoff when service has already experienced a significant degradation.

Therefore, in order to really exploit mesh network solutions with advanced services, it is necessary to provide an innovative architectural solution that offers:

- seamless mobility over multiple networks with local and wide coverage;
- optimized global node mobility to performs multiple heterogeneous networks in parallels,
- adequate support to advanced solutions for QoS.

In the framework of the European project EU-MESH [3], we are developing this kind of solution for mobile users in mesh networks, making a clean break from separate multimedia and data networks. Our solution leverages on a seamless handover at the application layer and selects optimal connection among heterogeneous networks with a cross-layering approach and specialized multimedia traffic handling. To support seamless and fast handoffs over heterogeneous networks and different operators, EU-MESH is exploring application layer solutions that exploit cross-layer monitoring, and optimize them for wireless mesh networks in terms of self-tuning and parameter adaptation, information collection, user interaction, and optimized handover decision procedures.

Cross-layer monitoring will enable effective adaptation of the delivered QoS to changing network conditions (due to operation anomalies, etc.). Furthermore, autonomic components based on proactive monitoring are used to self-optimise the internal parameters when the networks context changes, and to self-configure mobile clients (e.g., self-detect characteristics of the on-board hardware). In the future, this architecture will also use past experience to show anticipatory behaviours and/or learn about user preferences through statistical measures of trend.

This is obtained with a novel design approach based on autonomic components and cross-layer monitoring and control to optimize the performance of the WiOptiMo system [4], [5], [6], which provides seamless inter-network roaming by handling mobility at the application layer. As presented in Section IV, this architecture is empowered by an heuristic service classification approach (Section III) and a correspondent traffic classification algorithm (Section IV-A) for handling multimedia traffic at the network and the application layer, in order to better characterize the traffic.

II. WiOptiMo Overview

Wireless connections are highly variable and often unreliable; this is true for GPRS/EDGE or UMTS connections, but nowadays it happens frequently also in WLAN due to interference and mobility. A network connection can be often interrupted or lose quality during a journey (by train, by car, or also on foot). Interruptions are undesirable for traffic with QoS requirements, and are annoying in scenarios that require continuity, such as when we are transmitting traffic with QoS service requirements, but also, for example, connections to a secure server, which require a re-login after an interruption, or if the user is downloading a mail attachment from a mail server, or even more if a user is in the middle of a multi step transaction, such as buying market stocks or on-line shopping. Time and money can be wasted for this problem.

WiOptiMo [5] is a system that finds and sets up the best possible network connection in real-time according to a given set of metrics. The best connection can be chosen (in terms of bandwidth, reliability, security, cost effectiveness, or other local/global parameters), among all possible connections (of any kind) at a certain time and location. The choice of the best possible connection is done in a transparent, automatic/semi-automatic way without interrupting active network applications or sessions. If no switch is possible, the system hibernates the applications to re-establish the previous connection as soon as it becomes available again or to establish a new one (if the re-establishment time is greater than the application time out, the application may signal a network problem).

The architecture is composed by two main components: the Client Network Address and Port Translator (CNAPT) and the Server Network Address and Port Translator (SNAPT). These two components hide mobility to the client and the server. The CNAPT and the SNAPT jointly act as a middle-ware layer (Fig. 1), making the client believe to be running either on the same machine as the server or in a machine directly wired to the server (depending on the configuration adopted).

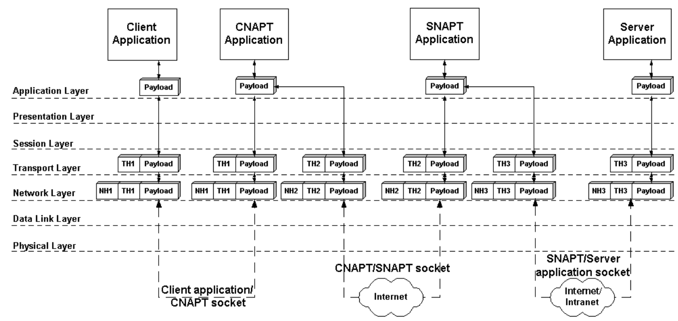

Fig. 1. The traditional unique socket between Client and Server application is substituted by three sockets, and the CNAPT and SNAPT components interface the client-server communication, so that Client and Server believe to be constantly and directly connected.

The CNAPT application acts as an application relay system and also launches a decision task in order to provide the client the best possible Internet connection in term of bandwidth, reliability and/or cost effectiveness. The decision task has two main activities: it continuously searches for new network providers and connectivity (Search Activity), and it continuously verifies the current Internet connection reliability and performance (Check Activity). One of the key aspect of WiOptiMo is that it is mainly implemented in C and JAVA, runs in the user space, and thus it is easily portable on each operating system, including palmtops and hand devices. The CNAPT is composed by a central core of less than 200 KB. The other elements of the CNAPT are satellites written in JAVA and C and their size vary depending on the platform of the device. The SNAPT is written in JAVA, and its size is about 90 KB. The software can work with any technology, and was tested with Ethernet (wired LAN), Wi-Fi (802.11b/g wireless LAN), GPRS (in Europe, U.S. and Canada) and EDGE in Italy. Within the framework of the EU-MESH project[1], we are completing the porting of this system in C, and modifying it to handle mobility and QoS requirements at the same time. A relevant aspect that motivated our design choice, is the fact that WiOptiMo is suitable to the requirements of personal devices: it can switch rapidly and with minimal overload, and is therefore suitable for real-time operations in multiple scenarios, ranging from entertainment up to business applications. The main challenge is to dynamically handle

[1]this work is partially funded under the European project 215320: EU-MESH, FP7 ICT-1-1.1: The Network of the Future

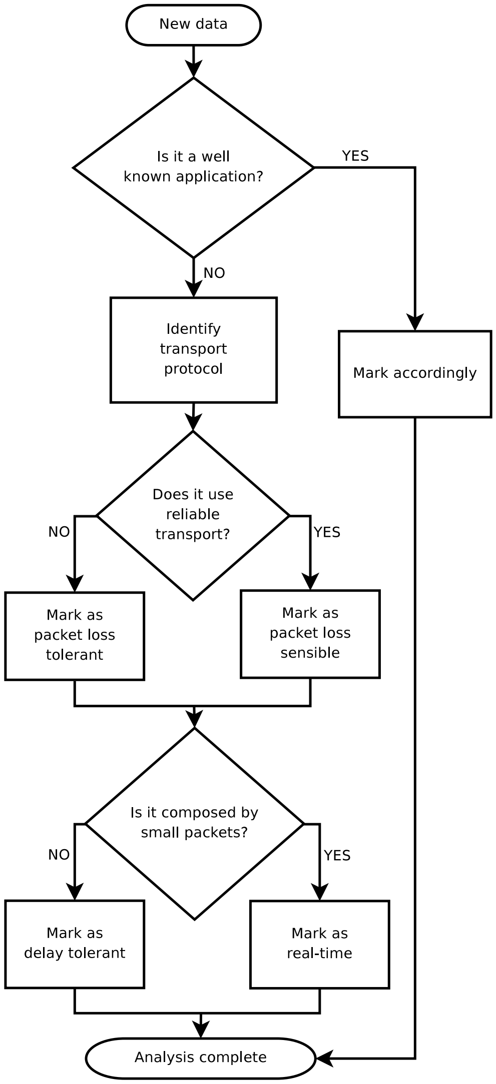

Fig. 2. The heuristic classification algorithm.

the QoS, which is fundamental to personal devices (as it allows to realize pervasive services). Furthermore, WiOptiMo operates at the application layer and it can use existing security solutions. It does not require the implementation of any custom security feature. If needed, the client or server byte streams between the CNAPT and the SNAPT can be compressed in order to reduce the amount of data exchange.

III. Heuristic services classification

The current usage patterns of the IP protocol are very heterogeneous. Most of the users download content from a Web server through the HTTP protocol. However, HTTP is often (ab-)used for bypassing restricted firewall policies and some ISPs rules. In fact, among those HTTP users, some data expose quite unexpected patterns or behaviours. For example, one of the most popular activities in the recent years is watching short video clips from a centralized server via capped bit rate HTTP sessions. This complicates the analysis of traffic when we simply base it on the traffic protocol, and we therefore propose another way for classifying it.

We start by identifying the *traffic characteristics* that impose strong constraints on the way this traffic is handled within the network.

A first classification of current multimedia traffic can be based on *traffic interactivity*. For example, another very common activity is messaging, both not interactive (e-mail) and interactive (XMPP and other messaging protocols). The first kind of traffic can be easily managed in the same way as HTTP traffic because a very little fraction of it requires quick feedbacks: once the content transmission is started, no user action can modify the traffic pattern (other than the user suddenly ending the session). Interactive traffic, however, needs different care because high latency degrades the user experience. Low latency is obviously a strong constrain also for real-time audio and video communications.

A second classification, orthogonal to interactivity, can be performed on the basis of *data loss tolerance*. Computer- to-computer communications are usually disrupted by incomplete or corrupted information, but human beings are usually able to reconstruct a significant part of missing data depending on the context. For example, in the case of few missing frames in a video stream, most users will barely notice the problem except if it involves a strong difference in the context, i.e. a scene change. For this reason real-time audio and video may be transmitted taking these capabilities into account.

Another parameter for traffic classification is the *application architecture*. In the 1990s almost all traffic was based on the client-server paradigm, but nowadays the popularity of peer-to-peer technologies has grown to account for 50% of resource utilization.

An exhaustive classification of network traffic based on the aforementioned features cannot be operated in a real-time mobility scenario. The analysis requires reading the entire packet payload, comparing it to known protocol schemes and checking for attempts to misuse common protocols.

For this purpose, we propose a **heuristic classification approach** that identifies some specific features within the header, instead of analysing the whole payload. The first step in the analysis is to determine the *traffic type* on IP networks based on the application protocol and port number. A large fraction of the traffic can be correctly classified with this first step.

The remainder of the traffic can be classified based on *network usage patterns*. A first element of this pattern is the *employed transport protocol*. If an application does not use a protocol with delivery guarantees, e.g. it uses UDP or DCCP, the traffic is expected to be tolerant to packet losses. Another element of the network usage pattern is the *average packet size*. If it is consistently smaller than the expected maximum size, it is very likely that the traffic has delay-sensitive content.

This heuristic analysis process is defined in the flowchart represented in Figure 2.

IV. Architecture improvements

A. Proposed gateway selection algorithm

The process described in the previous section can be implemented by accounting for two parameters: the *delay sensitivity* and the *loss tolerance*, while the *bandwidth requirements* are used for admission control and route selection.

Each type of communication has a *profile* in terms of delay sensitivity and loss tolerance, which are used to score the characteristics of the available network connections. For example, if the traffic requires very small delay but has a high loss tolerance, a network connection that can offer a quasi real time delivery, but has a non-negligible data loss rate is scored much higher than another one with zero losses, but high delay. The resulting list is then sorted and the solution with the higher score is chosen.

In several situations, the same gateway could be reached by multiple routes, and this can be fundamental for the admission control, as a secondary route may be promoted whenever the main one has not enough available bandwidth to satisfy the request. Therefore, the number of possible backup routes is also taken into account, as a secondary sorting metric, when multiple solutions with similar score are available. While the connection is active, the system monitors the network status and forces a route change if the current solution is no more suitable for the assigned type of traffic.

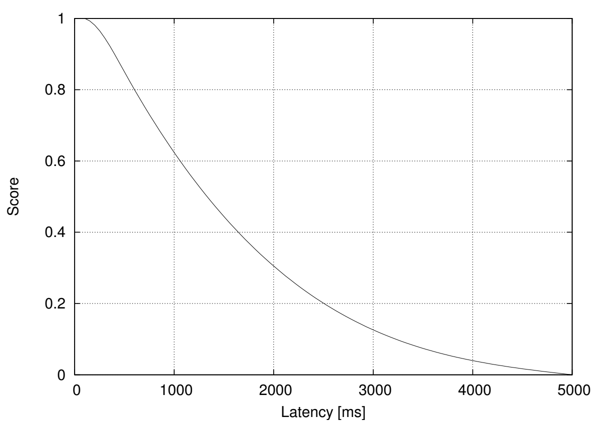

Fig. 3. Score for non interactive communication latency.

In the results reported by this paper, HTTP traffic and VoIP communications have been involved, so the scores for this kind of traffic has been determined as follow:

- Miller determined in [7] the latency threshold for human to computer interfaces, such as a web browser is. From this work, the parameters for a Gaussian distribution are calculated and the score for web site replies, file transfer start, or the establishment of any real-time session is reported in Figure 3.
- Human reaction to latency in voice communications has been heavily analysed by the International Telecommunication Union (ITU) and their results are available in [8]. The derived score is reported in Figure 4.

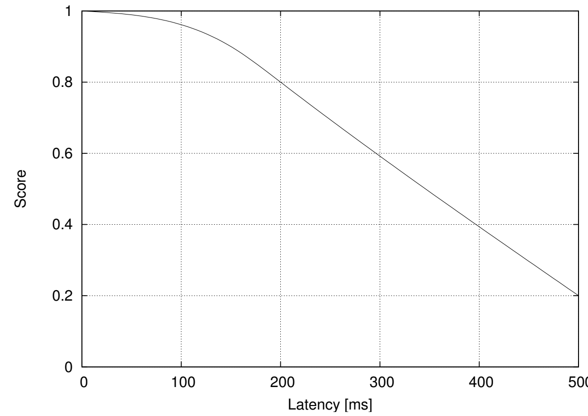

Fig. 4. Score for VoIP communication latency.

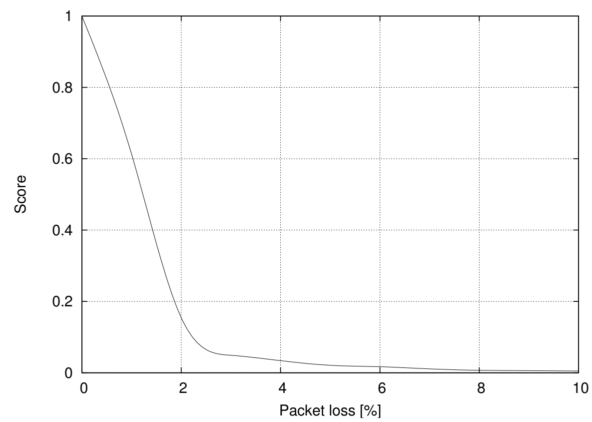

Fig. 5. Score for TCP traffic with simulated channel noise.

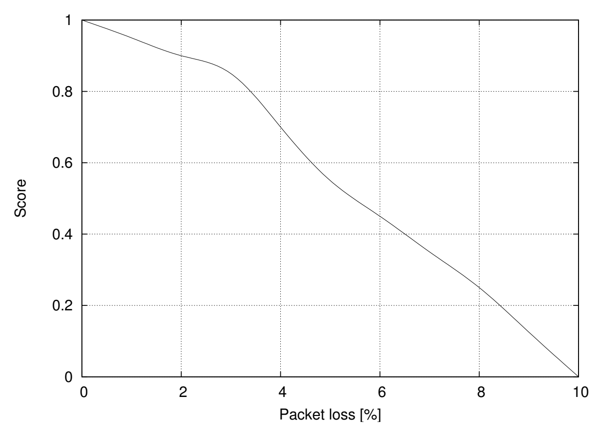

Fig. 6. Score for VoIP conversations with simulated channel noise.

- Web browsing and other non real-time activities are transported on the Internet using TCP. Its congestion control mechanisms assume packet losses as evidences of network congestion. The score reported in Figure 5 is derived by testing Linux TCP implementation on an emulated network link with the same packet loss scheme employed in Section V.
- Similarly the VoIP tolerance to packet losses is estimated using the average Mean Opinion Score (MOS) of Percep-

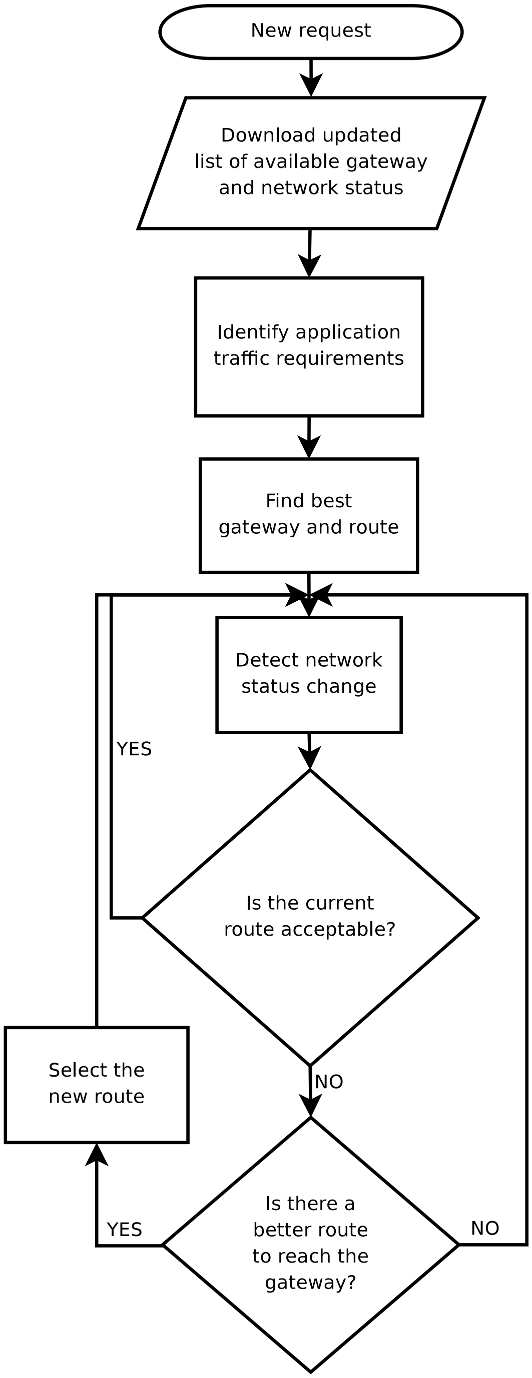

Fig. 7. The proposed gateway selection algorithm.

tual Analysis of Speech Quality (PESQ). The employed samples where encoded with G.726 at 40 kb/s. Figure 6 shows the results using the same error scheme employed in the experiments.

The resulting algorithm is presented in Figure 7.

B. QoS architecture overview

The mechanisms described in Section IV-A require the presence of multiple gateways to the destination, a network measurement system, and a traffic and admission control system. Multiple gateways are obviously needed to provide user devices with multiple opportunities to reach the destination. In a wireless mesh network, this condition is generally met. The measurement system is needed to determine the status of each gateway and path to the destination depending on the traffic type. This measurement should not be delegated to the user device because it would probably be inaccurate and incomplete. A traffic control system (TCS) is needed to accommodate new client requests with no QoS degradation for existing client.

As a client issues a new request, it begins by downloading a list of all available gateways and the respective network path information (load, latency, packet loss, etc.). The algorithm then chooses the most suitable path for the traffic type. The application can be identified with the help of the operating system, which for instance can provide the WiOptiMo client with information about which application has opened a given socket.

If the application has a well-known traffic pattern and a specific purpose, the choice can be operated easily. However, it is impossible to know the traffic type produced by any kind of existing software. In this case a default approach can be followed depending on the employed transport protocol: if UDP is used, the traffic is assumed to be loss tolerant and probably real-time, otherwise it is assumed to be delay tolerant, high bit-rate traffic. Meanwhile, the WiOptimo client can classify the data that is being transferred for future reference.

After that, the client sends its request to the TCS on the selected gateway, which checks if the requested resources can be provided in order to avoid saturation due to misbehaving clients. The TCS must also verify that the traffic offered to the gateway is compliant with the requested traffic profile. Non-compliant traffic must be shaped not to disturb to well-behaving clients. The TCS is also responsible for the selection of a new route to the assigned gateway whenever needed. The assigned gateway must remain the same to maintain the same front-end address even in the case of a client address change.

V. Experiments

To validate the proposed assignment scheme, we used part of the Linux mac80211 stack to emulate a wireless mesh network with ten clients connected to the same node. Each of them may randomly request an HTTP download or perform a VoIP call, with up to five requests per each traffic type.

The network, as illustrated in figure 8, provides two gateways. The first one is reachable without packet loss with a latency of approximatively 250ms while the second is reachable within 45ms but packets may be dropped due to channel errors, which are simulated by a two stage Markov chain in order to generate bursts of packet loss. The transition probability from the good state and the bad state is 0.01, while the reverse transition probability is 0.5.

The ten clients are hence assigned to the two available gateways depending on the algorithm in Section IV-A and, as a performance baseline we employ a random gateway assignment. Average results are reported in Table I.

	Random assignment						Proposed assignment					
Type of Traffic	Avg. Through-put [Mb/s]	Avg. Worst Through-put [Mb/s]	Avg. Latency [ms]	Avg. Worst Latency [ms]	Avg. MOS	Avg. Worst MOS	Avg. Through-put [Mb/s]	Avg. Worst Through-put [Mb/s]	Avg. Latency [ms]	Avg. Worst Latency [ms]	Avg. MOS	Avg. Worst MOS
HTTP download	6.141	**0.524**	179	291	-	-	4.743	**3.871**	298	384	-	-
Voice over IP	0.039	0.037	**181**	**299**	3.73	3.17	0.037	0.037	**56**	**61**	3.36	3.16

TABLE I
SIMULATION RESULTS.

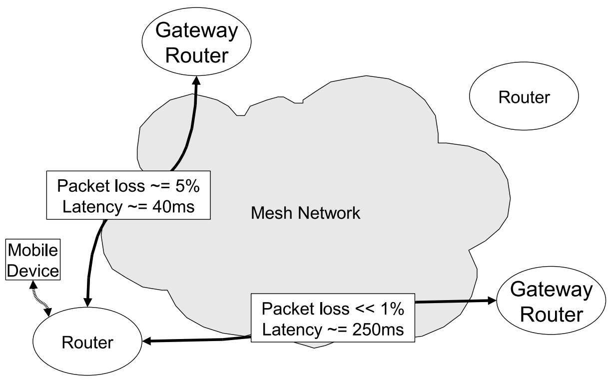

Fig. 8. Simulated network topology.

As we can see in the table, the average throughput is higher for the random assignment, due to the fact that with the latter a given client is equally likely to employ either gateway. Where we can see real improvement is the average worst case: the throughput is close to the average value while it really drops for the random allocation. This is due to the fact that, with our proposed classification, we are able to identify the traffic and to treat it appropriately.

VI. Conclusions

In this paper we presented an architectural solution for handling multimedia traffic with QoS requirements in mobile and wireless networks. The solution is based on an heuristic approach for service classification and on a classification algorithm implemented with a network measurement system and a traffic and admission control system. This was simulated on a Linux platform with encouraging preliminary results. The most interesting achievement is the fact that we have been able to identify a large part of the traffic and, consequently, to handle it appropriately. This is confirmed by the fact that the average throughput does not vary too much in the worst case. This confirms the validity of the proposed approach, and paves the way to the study of more dedicated shaping and policing strategies to be employed in a real mesh network. As the proposed architectural solution provides a seamless handover between heterogeneous networks, it can offer support for multimedia applications for low and discontinuous rate networks as the wireless and mobile networks. This solution enables the integration of data and multimedia networks for the support of advanced IP multimedia services. At present, this requires a large investment in infrastructures, as well as a significant management overhead. This represents a high entry barrier for SMEs, start-ups, and individual entrepreneurs. This also inhibits the emergence of new services. That are vital to the business case for investing in advanced networking technologies. Finally, this solution would speed up the network development, easing the delivery of emergency solutions, the coverage of rural areas or the entering of new countries.

References

[1] IEEE 802.11 working group, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification/amendment 2: Higherspeed physical layer (PHY) in the 2.4 GHz band," *IEEE Standard*, November 2001.

[2] A. Balachandran, G. Voelker, and P. Bahl, "Wireless Hotspots: Current Challenges and Future Directions," *Proceedings of 1st ACM Workshop on Wireless mobile applications and services on WLAN hotspots (WMASH'03)*, September 2003.

[3] Eu-mesh. [Online]. Available: http://www.eu-mesh.eu/

[4] S. Giordano, D. Lenzarini, A. Puiatti, and S. Vanini, "WiSwitch: Seamless Handover between Multi-Provider Networks," *Proceedings of the 2nd Annual Conference on Wireless On demand Network Systems and Services (WONS)*, January 2005.

[5] G. A. D. Caro, S. Giordano, M. Kulig, D. Lenzarini, A. Puiatti, F. Schwitter, and S. Vanini, "WiOptimo: A Cross-Layering and Autonomic Approach to Optimized Internetwork Roaming," *AHSWN journal*, May 2007.

[6] S. Giordano, M. Kulig, D. Lenzarini, A. Puiatti, F. Schwitter, and S. Vanini, "Method and system for a predictive detection of the wireless communication channel degradation when the channel is used only or prevalently for constant-bit-rate traffic," *patent pending*, May 2006.

[7] R. B. Miller, "Response time in man-computer conversational transactions," in *Proc. AFIPS Fall Joint Computer Conference*, vol. 33, 1968, pp. 267–277.

[8] "ITU-T recommendation G.114," International Telecommunication Union, Tech. Rep., 1993.