EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Eur. Trans. Telecomms. (in press)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/ett.1241

Communication Networks

IntServ6: an approach to support QoS over IPv6 wired and wireless networks

Jhon J. Padilla[1,2]* and Josep Paradells[2]

[1]*Pontificia Bolivariana University, Colombia*
[2]*Technical University of Catalonia, Spain*

SUMMARY

In this paper we propose a new approach for Quality of Service (QoS) support on Internet. This approach, named IntServ6, is based on the Integrated Services Architecture (ISA). It takes advantage of the IPv6 header flow label field to improve a set of the standard ISA properties such as reservations within tunnels, flows aggregation and interconnection with MPLS transport networks. IntServ6 can be used for QoS support in IPv6 wired and wireless networks. This paper describes the IntServ6 operation and performance evaluation over both environments. Evaluation results show that this approach has a better router performance with respect to the standard IntServ. Thus, IntServ6 reduces the mean packet delay and reduces the packet delay dependence with the mobility. Copyright © 2007 John Wiley & Sons, Ltd.

1. INTRODUCTION

Next generation networks will be based on IPv6 protocol. As a result, these networks will have issues that the standard IPv4 cannot support such as a great IP addresses number, users mobility, Quality of Service (QoS), security, etc. Such networks will have to support several services classes for elastic applications and real time applications. Under this environment, the QoS support capability is an important issue for these networks. Nowadays, to support QoS on the Internet there are mainly two standard architectures: Integrated Services Architecture (ISA) and Differentiated Services Architecture (DSA). ISA can be used on access networks since it easily adapts to the user's requirements. However, it has scalability problems. On the other side, DSA is very scalable but it cannot easily fit the final user's requirements. Therefore, it is useful for transport networks. On a third side, the Multi Protocol Label Switching (MPLS) supports traffic engineering principles. As a result, MPLS can be used either with ISA or DSA in order to support QoS in a better way on the Internet.

Our approach, named IntServ6 is based on Integrated Services but with an important difference in the classification process: while standard ISA (also named IntServ) uses the five-tuple for the packet classification, IntServ6 uses the IPv6 flow label field to identify the flow a packet belongs to. This idea is not new and it is the original goal of the flow label field within the IPv6 protocol header. However, this field has not been used for this purpose. Besides, the flow label field utilisation has not been completely standardised. Although current standards describe general recommendations [1, 2], they are not clear yet about the specific use of this field. The use of the flow label field to classify the packets on IntServ6 improves the router performance and several issues in which standard IntServ currently has solutions with some disadvantages. These issues include flows aggregation, reservations within tunnels and interconnection with transport networks.

* Correspondence to: Jhon J. Padilla, Universidad Pontificia Bolivariana Ingeniería Electrónica, Km 7 Via Piedecuesta, Bucaramanga, Colombia.
E-mail: jpadilla@upbbga.edu.co

Received 9 May 2006
Revised 25 October 2006
Accepted 26 April 2007

In order to demonstrate the benefits of our approach, we have developed some models to compare IntServ6 with the standard IntServ and we have obtained important results. A first model was developed in Reference [3]. It describes a router performance model with parameters such as the required number of threads and the memory bandwidth. Besides, a second model detailed in Reference [4], describes the mean packet delay dependence with mobility parameters such as the new calls and the handoff calls traffic intensity over Integrated Services wireless access routers (AR). Now, in this paper, we introduce a general model for the mean packet delay, which is valid for Integrated Services wired and wireless routers.

To summarise, we have obtained a proposal that supports QoS in both wireless and wired networks. In order to explain our approach, Section 2 describes and compares the classification process of standard IntServ and our approach, IntServ6. In Section 3, we explain the reservations within tunnels problem and its solution for IntServ and IntServ6. Section 4 describes the flows aggregation problem on IntServ and how our approach performs and improves this issue. Section 5 describes the interconnection of IntServ6 networks and MPLS transport networks. Section 6 has a description about the reservations mobility and the extensions of IntServ6 to solve this problem. Section 7 presents our mathematical models and Section 8 shows our simulation results. Finally, Section 9 presents our conclusions.

2. RESERVATIONS SUPPORT WITH THE FLOW LABEL FIELD

IntServ6 is based on the IntServ architecture for QoS support. In order to improve IntServ routers performance, IntServ6 uses IPv6 flow label to speed up the process of packet classification on routers. The packet classification process of an IntServ router uses five fields of the received IP packets (Source IP address, Destination IP address, Source port, Destination port and Protocol Identification), commonly known as the five-tuple, to obtain a hash number. This number is used to search a pointer to the resource reservation table in a hash table; the resource reservation table is used to hold scheduling information. Each time a packet arrives at any IntServ router over the reservation path [5, 6], the packet classification process is carried out and the flow searching on the hash table is performed with a matching scheme.

On the other hand, on IntServ6 the hash number calculation is performed at source host and not at the routers. Thus, while reservation setup is done, the hash number is carried to each router on the flow path using PATH and RESV messages of the RSVP protocol. This action is necessary to create a new entry to the hash table. On the other side, during normal data packet forwarding (see Figure 1) each IntServ6 router extracts the flow label field from IPv6 packets and it uses this field as a searching index to find reservation parameters (by using a tagging

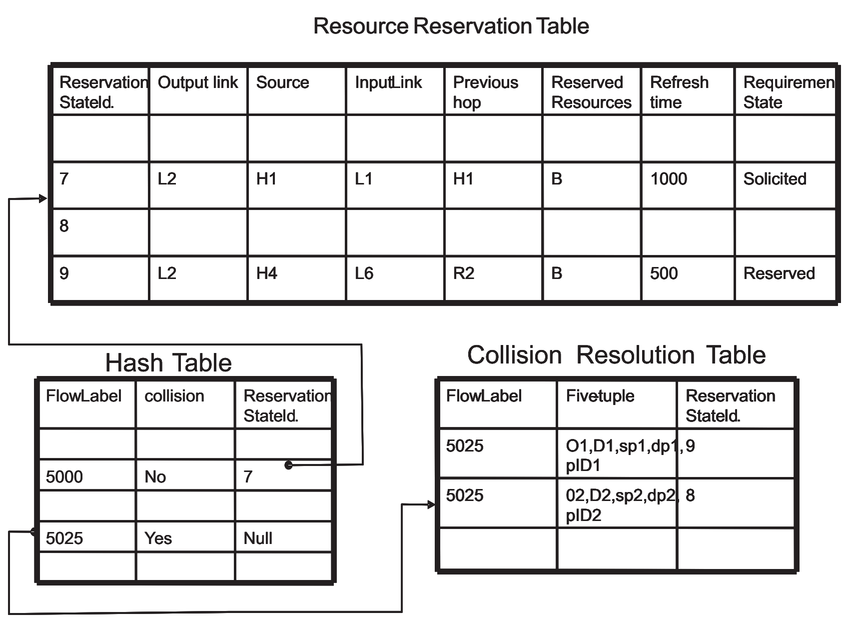


Figure 1. Reservations table management in IntServ6.

DOI:10.1002/ett

scheme) and the next forwarding link in the resource reservation table. This is an important difference because in the hash table searching process IntServ routers use a matching search scheme while IntServ6 routers use a tagging search scheme (see References [3, 7, 8]). The use of the flow label field was described in References [1, 2], and our proposal was created according to them. The hash number calculation at the source host is feasible and does not represent a problem because the resulting hash number is the same independent of the calculation site (the five-tuple and the hash key calculation algorithm are the same).

Nevertheless, there is an additional problem due to the use of the hash number: a collision may occur. A collision occurs when a hash number calculated at the Host was already assigned to another flow at the router. To solve this problem, it is necessary to use a collision resolution table (see Figure 1), which holds information about collided flows.

On IntServ6, this table has fields such as the flow label, the five-tuple, and a resource reservation entry pointer. When a collision occurs (this is, when the collision field into the hash table is on 'yes' state), the router searches an entry for a flow label and a five-tuple matching in the collision resolution table. After that, the router extracts the resource reservation table pointer. This process is similar for both technologies, IntServ and IntServ6, but IntServ routers do not have the flow label field on the collision resolution table.

3. RESERVATIONS WITHIN TUNNELS

One important aspect to resolve in IP tunnelling is to maintain resource reservations of external flows within the tunnels. When standard IntServ is used for QoS support, a problem arises due to the hash number calculation for each packet arriving. Within the tunnel, the five-tuple is hidden and it cannot be read from the external IP packet header. The solution for standard IntServ is described in Reference [9]. Thus, to distinguish between several reservations within the tunnel, it is necessary in that approach [9] to transport the external IP packets within UDP tunnel packets. As a result, one UDP port for each resource reservation is assigned. Then, that approach resolves one network layer problem at the transport layer level.

On the other hand, our approach gives a simple solution to the hidden five-tuple problem within the tunnels. Besides, it resolves the problem at the network layer level. In IntServ6 (see Figure 2), the IPv6 flow label field is used to distinguish resource reservations. Thus, when a packet is encapsulated at the Tunnel Input router, the original flow label remains into the original IPv6 packet header. However, a new flow label is assigned to each external packet arriving to the tunnel. This is done using the same hash key calculation principles mentioned above. This new label is named tunnel label. It is carried on the external IPv6 packet header and it identifies the tunnel resource reservation. In addition, the tunnel resource reservation should be established before the forwarding IP packets starts. Also, the tunnel resource reservation should be sufficient to maintain reservations for the arriving flows. The method to calculate the required number of resources within the tunnel is not explained in this document.

In order to achieve a better process understanding, first we should define two reservation types (see Figure 2): The End to End (E2E) reservations, which are resource reservations allocated before and after the tunnel and the

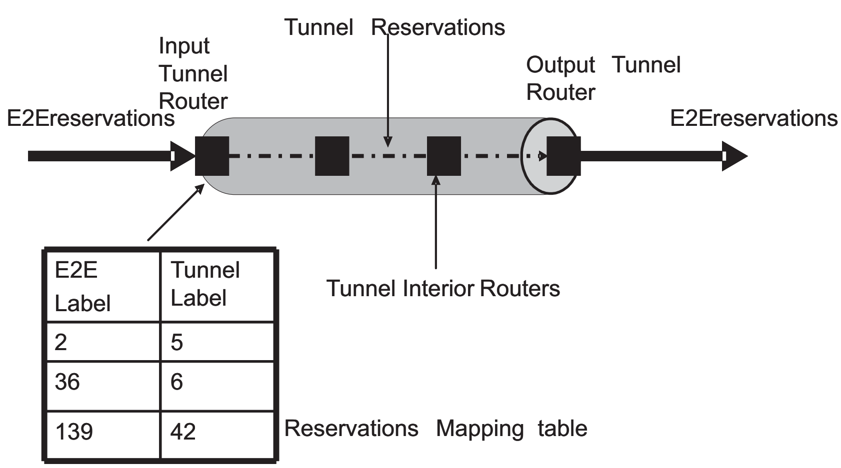


Figure 2. Reservations within tunnels with IntServ6.

Tunnel Reservations, which are resource reservations allocated over all the routers within the tunnel. When a tunnel must be established, PATH and RESV messages should be interchanged between the Tunnel Input router and the Tunnel Output router. The Tunnel Input router specifies the required resources by means of the SenderTSpec RSVP object into a PATH message, and the Tunnel Output router should decide exactly how many resources it requires. Then, it sends this information to the Tunnel Input router by means of a RESV message. When a reservation is established within the tunnel, a new entry is added to the reservations mapping table. This table is allocated at the Tunnel Input router. Besides, a new entry should be added in the resource reservation table and in the hash table at each router over the tunnel path. Each entry into the reservations mapping table has the E2E label and its correspondent tunnel label. At this moment, the tunnel path is ready for packet forwarding.

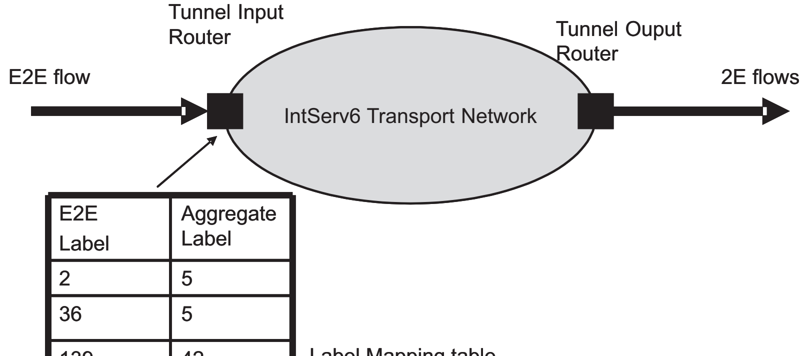
On the other hand, the PATH and RESV messages for the E2E Reservations travel into the tunnel as encapsulated messages. Therefore, these messages are not recognised by the Tunnel Interior routers, whereas PATH and RESV messages of the Tunnel Reservation travel without the encapsulating process and they are identified by the Tunnel Interior routers.

4. FLOWS AGGREGATION

Flows aggregation on IP networks allows a hierarchical organisation of the Internet. Thus, high level routers switch packets by destination domain address rather than by destination IP address. This issue speeds up the packet switching process on these routers. Besides, it is possible to reduce the routing table size by means of flow aggregation. Then, a solution to support QoS should easily fit the Internet flows aggregation issue.

A solution given for the standard IntServ is described in Reference [10], where to support flows aggregation it is necessary to use a Differentiated Services transport network. Under this approach, an Aggregation Edge Router and a Disaggregation Edge Router should be used to map the E2E IntServ reservations into Differentiated Services classes of the transport network. This is, the same standard IntServ does not have this capability and should use another technology to supply this issue. The problem to support flow aggregation on IntServ is the classification procedure because a router should perform the five-tuple calculation for each arriving packet. Then, it is necessary to use all the destination IP address bits to support the resource reservation.

On the other side, IntServ6 allows flows aggregation based on an MPLS label stacking similar method. This process can be observed on Figure 3. The basic principle is the use of tunnelling. When a higher number of flows travel from the same source network domain to the same destination network domain, an IntServ6 based transport network can be used in an easy way to carry the traffic aggregate. To do this, it is necessary that an Input Edge router on the IntServ6 transport network puts all the packets belonging to this aggregate in a tunnel between itself and the output edge router. Thus, all the flows belonging to an aggregate should have the same flow label number into the external IPv6 header. As a result, each Input Edge router on the IntServ6 transport network should have a Label Mapping Table, which holds entries with the E2E flow label number and the correspondent Aggregate flow label number into the transport network. Moreover, the



E2E Label	Aggregate Label
2	5
36	5
139	42

Figure 3. Flows aggregation with IntServ6.

DOI:10.1002/ett

Table 1. Simulation parameters.

IXP 1200 Network Processor Parameters	
	Value
Parameter	
L_{sdram} (sdram reading time)	160 ns
L_{sram} (sram reading time)	80 ns
Word processor size	32 bits
Processing times	
$t_i = 1L_{sram}$; (hash table verification time- IntServ, IntServ6)	80 ns
$t_e = 2L_{sram} + 32L_{sdram}$; (packet routing time, is the same for IntServ and IntServ6)	5280 ns
$t_{comp} = 10 * L_{sram}$; (five-tuple comparison time for collision resolution process)	800 ns
$t_q = 10 * L_{sram}$; (five-tuple reading time for hash table searching)	800 ns

Output Edge router should only extract E2E IP packets from the tunnel and then forward them.

Thus, in Figure 3 we can see that an aggregate is a set of external flows that uses the same tunnel label. This is, the same tunnel reservation is established for one or more flows. Then, as several E2E flows are mapped to the same tunnel label, the Tunnel Output router should calculate the number of tunnel resources required to maintain the same QoS for all E2E flows within the aggregate. The method to calculate the required number of resources within the aggregate is not explained in this document.

In addition, under this environment a new flow could appear or an existent flow could disappear into the aggregate. To solve this situation, IntServ6 introduces two new signalling RSVP messages. When a new flow appears, a PATH message with a RESIZE object is sent from the Tunnel Input router to the Tunnel Output router. This message carries the Aggregate flow label field, the additional required resources and a SIGN bit on SET state. The SIGN bit indicates to the path routers that the resources should be added to the specified aggregate flow whenever the flow is accepted.

The Tunnel Output router determines if these new resources can be added and confirmed by means of a RESV message with the same RESIZE object, which travels over the tunnel path up to the Tunnel Input router. When the resource addition is confirmed, the Tunnel Input router allocates a new entry on the reservations mapping table and modifies all the other necessary tables mentioned above. When an existent flow disappears, the Tunnel Input router or the Tunnel Output router should send a PATH or RESV message respectively with the same RESIZE object, specifying which resources should be reduced, and the E2E flow label that should be extracted from the reservations mapping table at the Tunnel Input router (the last one is for RESV messages only). Besides, the SIGN bit is on RESET state to inform that these resources should be eliminated from the tables.

5. INTERCONNECTION WITH TRANSPORT NETWORKS

Nowadays, interconnection with MPLS transport networks is an important issue of IP networks. Thus, any QoS support architecture should provide solutions for this aspect. IntServ6 allows easy interconnection with MPLS networks. As MPLS networks provide some resources to the users, and these resources should be compliant with a pre-defined Service Level Agreement (SLA), any on-line negotiation between the user networks and the MPLS transport networks is not necessary. Hence, an IntServ6 client network only sends its packets to the MPLS transport network, and the MPLS Label Edge router (LER) performs the same standard procedures that should be executed with any IP network. This is, the MPLS LER should read the IPv6 packet header and should give them the SLA pre-defined treatment.

However, IntServ6 issues can be used to improve the interconnection with an MPLS transport network. If we suppose that the MPLS network is based on Integrated Services for QoS support and that it uses the RSVP-TE protocol to perform label distribution, we can obtain an environment similar to that in Figure 4. Under this environment, IntServ6 reservations can be mapped on the MPLS network by means of two reservations mapping tables allocated into the MPLS LER and the IntServ6 Edge router respectively. The table allocated at the MPLS LER holds both the IntServ6 Input Label and the MPLS Input Label used within the MPLS network. Thus, when an IntServ6 packet arrives to the MPLS transport network, the MPLS LER reads the IPv6 flow label field and searches this label on the reservations mapping table to obtain the MPLS/label space assigned to it. To add new reservations into the reservations mapping tables, the IntServ6 Edge router should request an IntServ6 Input Label to the MPLS LER. This is done by means of a PATH message, which contains the required resources for the new flow. The MPLS LER performs the MPLS internal label allocation by means of RSVP-TE procedures. Then, the MPLS LER informs to the IntServ6 Edge router which IntServ6 Input Label should be used for this flow (by means of a

DOI:10.1002/ett

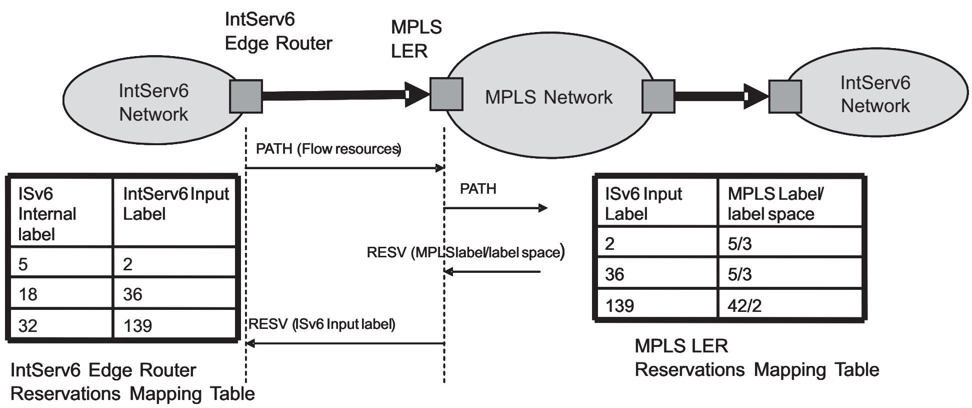


Figure 4. Automatic interconnection between IntServ6 and MPLS networks.

RESV message). In consequence, the IntServ6 Edge router adds a new entry within its reservations mapping table, which holds the flow label internally used on the IntServ6 network and the IntServ6 Input Label assigned for the MPLS LER. This procedure was designed to avoid that MPLS LERs have to resolve hash key collisions. Thus, this function is performed only at the IntServ6 Edge routers.

This environment speeds up the MPLS LER packet forwarding operation because it does not have to read the IP address the standard way. Then, the MPLS LER only has to read the IPv6 flow label field and then, it forwards the packet over the MPLS network. In addition, by means of this configuration, an MPLS transport network can fit its internal resources to its user requirements in an automatic way that easily changes resource reservations in time. In the current way, over an MPLS network, the resources assigned to the users are always the same in compliance with the SLA.

6. EXTENSIONS FOR RESERVATIONS MOBILITY SUPPORT

Next generation mobile networks will be based on IPv6. Also, they will support services with elastic traffic, such as file transfer, and real time services such as voice and video. IntServ6 has extensions for these mobile environments, so it can support QoS over mobile IPv6 based networks. These extensions were introduced recently in Reference [11]. The main problem to solve reservations support over IntServ6 mobile networks is that the Care of Address (CoA) changes when a handoff occurs. As a result, the five-tuple changes and then, the flow label number used to identify the flow reservation changes as well. As a consequence, the resource reservation fails. To avoid this problem, IntServ6 extensions for mobile environments should establish a new reservation over the new packets path and then, delete the path state for the old reservation.

However, another problem arises with the handoff latency because it increases due to the additional procedures required to establish the new reservation and delete the old reservation. To reduce the handoff latency, IntServ6 could operate under a Hierarchical Mobile IPv6 (HMIPv6) network [12]. With the use of HMIPv6, the changed part of the packets path is reduced to only a few routers closely located to the mobile terminal. Thus, the new reservation establishment is reduced to a small changed path segment. Then, the reservation handoff latency is reduced. Another solution that can reduce handoff latency even more is through the fast handoff protocol [13].

As a result, our network architecture (see Figure 5) uses HMIPv6 for mobility support. Another important issue of our solution is that it makes minimum modifications to the standard mobility protocols. Thus, to support reservations mobility, our approach is based on a solution proposed by Paskalis *et al*. [14] but with some modifications.

Our approach uses HMIP Mobility Agents, which supports the HMIPv6 protocol, and Reservations Mobility Agents (RMA) allocated at all the Mobility Anchor Points (MAPs) of the network. These RMAs are alerted about reservations mobility events by the HMIP Mobility agents allocated at the MAPs. When an RMA receives a Mobility Event Alert, it begins a reservation handoff procedure. Several handoff situations could occur. A first situation is the local handoff, which only involves path changes at the AR level but without MAP changes. A second situation is

DOI:10.1002/ett

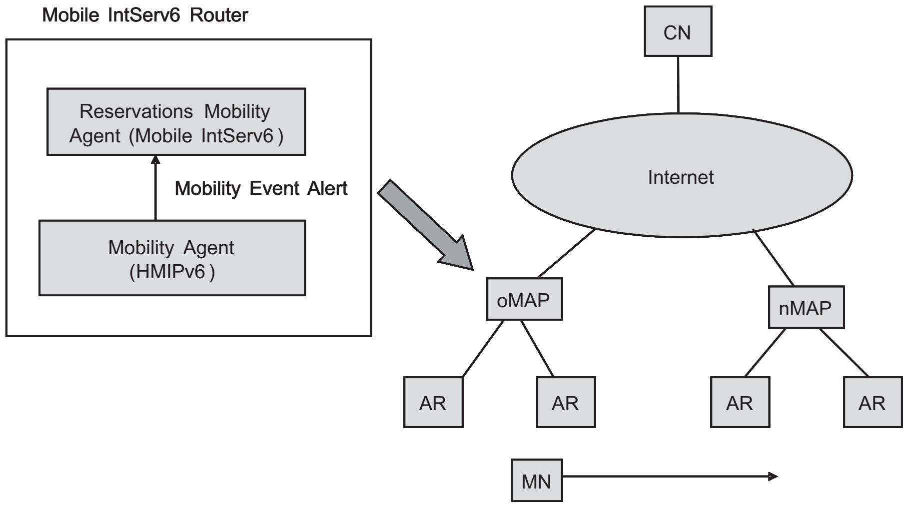


Figure 5. Wireless network architecture for IntServ6.

the inter-domain handoff; this is, a handoff could occur between ARs belonging to different MAPs. The main situations are described in Figures 6, and 7(a, b), where a local handoff and an inter-domain handoff are described, respectively. In order to give a simple explanation, we only describe a local handoff with the Correspondent Node as the data source, and the Mobile Node as the data destination.

As we can see in Figure 6, when a local handoff succeeds, the MAP Mobility Agent receives a Binding Update (BU) message from the MN with the new LCoA. At this moment, the MAP Mobility Agent sends a BU acknowledgement to the MN while the MAP mobility Agent informs to the RMA about a mobility event.

In consequence, the RMA initiates a new reservation establishment and it sends a PATH message to set up the

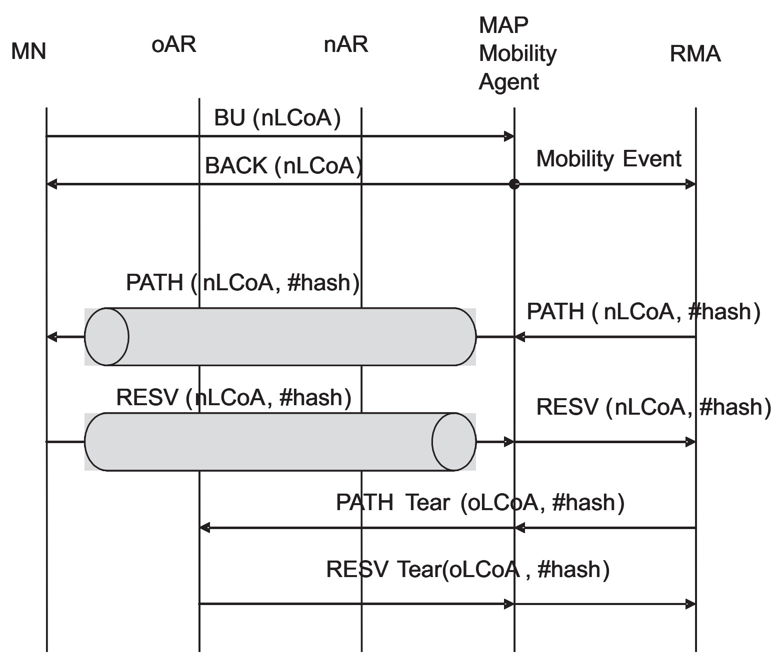


Figure 6. Reservations local handoff process on wireless IntServ6 networks.

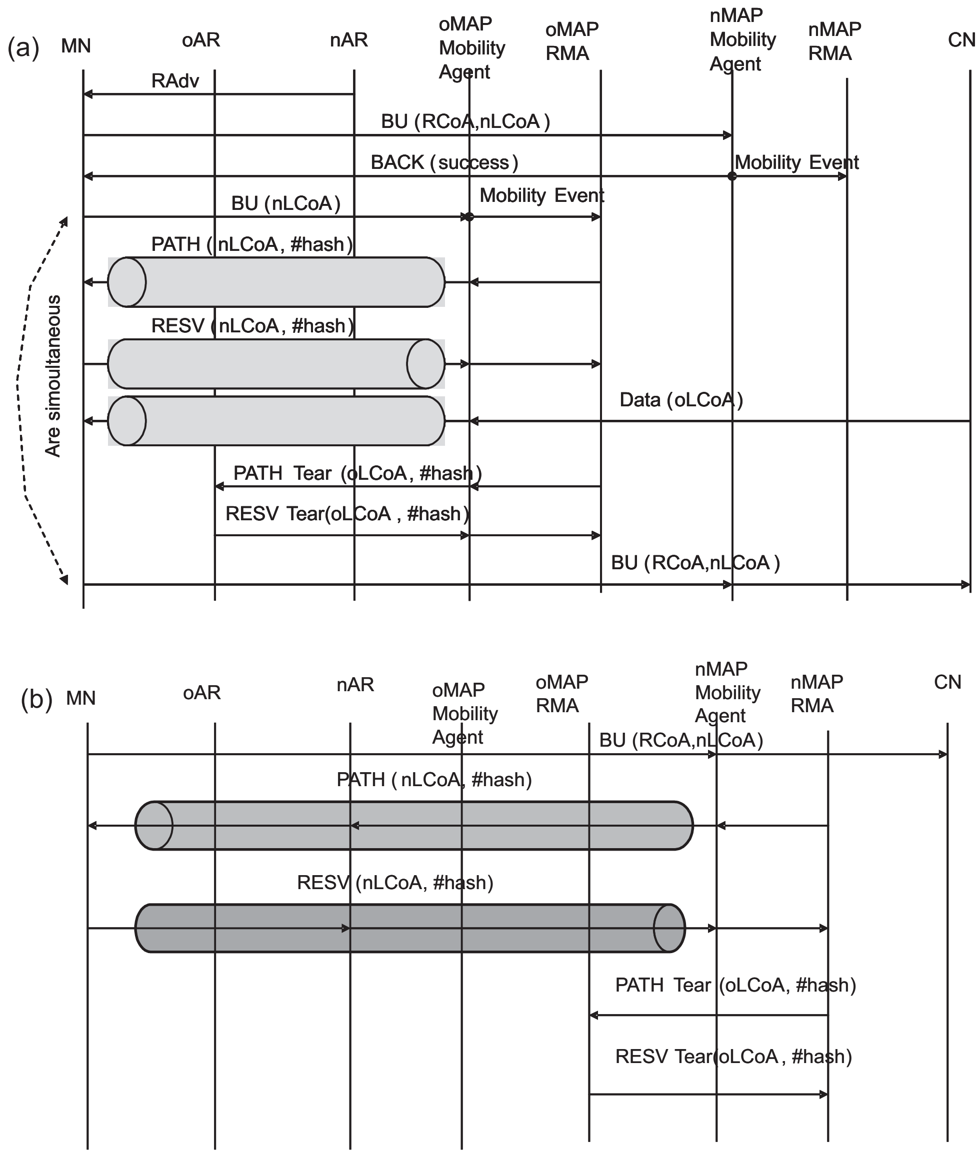


Figure 7. Reservations inter-domain handoff process on wireless IntServ6 networks.

new reservation up to the MN. Once the new reservation is established, the old reservation is dropped by means of PATH Tear and RESV Tear messages between the RMA and the old access router (oAR). The tunnel reservation set up is not shown in these figures because for explanation simplicity we assume that it is implicitly performed when the tunnel is established between the MAP and the MN. The tunnel reservation process follows the rules explained above in Section 3.

7. INTSERV6 MATHEMATICAL MODELS

A first IntServ6 evaluation was performed as in Reference [3]. This evaluation developed a router architecture model and it obtained parameters such as Memory Bandwidth, Threads Number and Throughput for the standard IntServ and for IntServ6. In that paper, a better router performance for IntServ6 with respect to IntServ was demonstrated.

Now, in order to evaluate IntServ6 at a QoS level, we started from the fact that the main QoS parameter for Integrated Services is the packet delay [15]. Thus, we developed a mean packet delay general model that can be used in wired routers and wireless access routers. The wireless access router case was introduced in Reference [4]. The general model will be explained as follows.

7.1. General model for an Integrated Services Router

An Integrated Services Router (see Figure 8) is composed mainly of two parts (a similar model is described in Reference [6] for any IntServ router over the reservation path). The first part is the reservations manager, which is composed of the admission control scheme and the reservations setup agent. The second part performs the packet forwarding process. It is composed of the policies control module, the classification process and the scheduling process. As we can see in Figure 8, these two main parts are related across the Flows reservation table. Then, when a new call is accepted, the number of flows on the Flows reservation table will increase in one flow. Thus, when a packet arrives, it should be classified, i.e. it should be found on the Flows reservations table. Finally, it should be scheduled over the correspondent output link. In the following sections, we will explain the mathematical models developed for the packet forwarding process over Integrated Services Routers (Subsection 7.2) and then, we will show how to introduce the mobility effect for wireless access routers (Subsection 7.3). The obtained models can be used to evaluate the standard IntServ and the IntServ6 routers performance. The main difference between both technologies, at a router architecture level, is given by the classification process. In the next section we will explain this aspect in a more detailed way.

7.2. Packet forwarding model for an integrated services routers

We started from a model of the packet forwarding process of an Integrated Services Router. This process is composed of two subsystems: the classification process followed by the scheduling process. Next, we will describe each of the mentioned parts of the router.

7.2.1. Scheduler model

The stochastic model of the scheduling process was extracted from a model developed by Zhang *et al*. [16]. Zhang's approach obtained a model for a specific i session of a *PGPS* scheduler as M/G/1 tail and obtained expressions for mean time delay, variance and mean delay upper bound. This model was also particularised for an M/M/1 tail with the same parameters mentioned above. We have used this M/M/1 tail for our modelling and the expression used for the upper delay bound is,

$$\bar{d}_i \leqslant \frac{\bar{L}_i}{w_i C - \lambda_i \bar{L}_i} \tag{1}$$

where $\bar{d}_i$ is the mean packet delay for the i session, $\bar{L}_i$ is the average packet length for session i, C is the total output server capacity (in bits per second), w_i is the flow weight and λ_i is the arrival average rate for the i session.

7.2.2. Classifier model

On the other hand, we have obtained a stochastic model for the classifier mean packet delay, which is based on the States Transition Diagram (STD) of the process

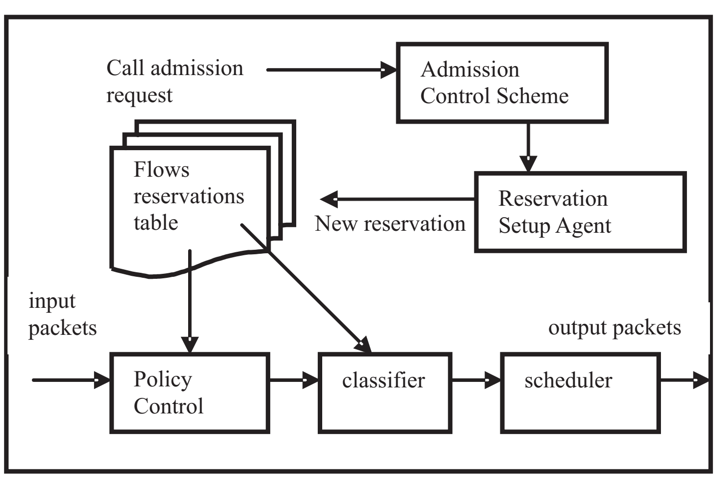


Figure 8. Integrated Services router model.

DOI:10.1002/ett

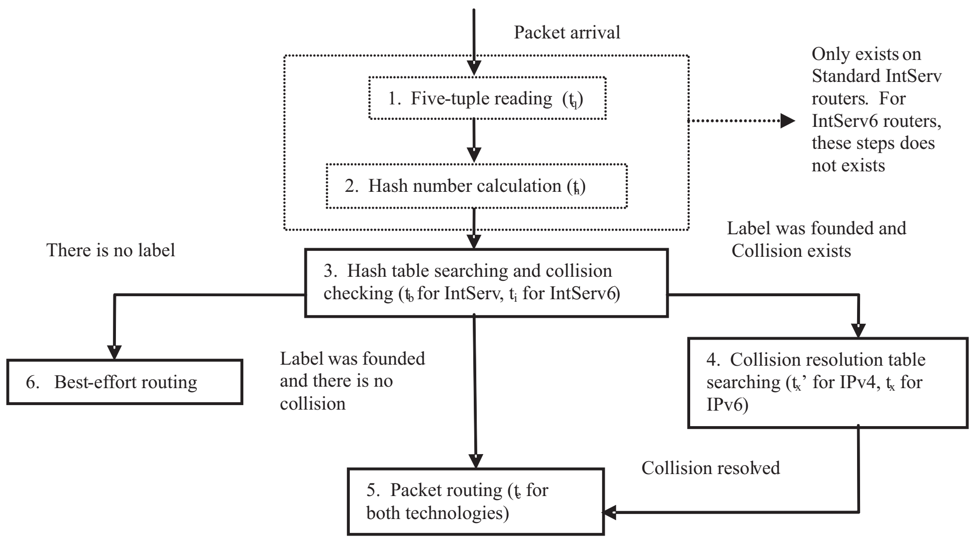


Figure 9. State transition diagram for the classification process on Integrated Services.

(see Figure 9). This STD changes for each technology (*IntServ* and *IntServ6*). In this way, the expressions obtained change for both approaches. In Figure 9 we can see that states 1 and 2 only exist on standard *IntServ* routers (for *IntServ6* routers, these steps do not exist). Besides, state 3 is different for both technologies because IntServ uses a matching scheme while *IntServ6* uses a tagging scheme. The method to obtain an expression for the classifier packet delay is to assign a time delay for each state (expressed in parenthesis in Figure 9). Then, to obtain an equation that takes into account the states probability, we calculate the product of each time parameter with its correspondent probability. Most states have a constant delay calculated from the sequence followed for each specific process. However, there are two processes which have a random delay. First, the *hash table* searching process for standard *IntServ* because it uses a matching scheme (as described in References [7, 8]); then the matching search has a random behaviour with a geometric distribution because each attempt represents a reading of the table, the result could succeed or fail, and this test is done until a successful event occurs. For *IntServ6* this process is deterministic because it uses a tagging scheme, and it only takes one hash table reading for this technology. A second random process is the collision resolution table searching process, which has a matching scheme (and a geometric distribution too) and it is similar in both technologies, *IntServ* and *IntServ6*. The service time of this process changes for *IPv4* and *IPv6* due to different header format (t_x and t'_x for the collision resolution process using *IPv6* and *IPv4*, respectively); as we use the *IPv6* header format for our study, then we will use t_x for the expressions obtained.

After that, we calculate the service time for the classification process for each technology. Then, we add each step time delay for each case. Each time delay parameter is defined in Figure 9. Thus, the expression for service time on IntServ is,

$$x_{\mathrm{co}} = t_{\mathrm{q}} + t_{\mathrm{h}} + (t_{\mathrm{b}} + t_{\mathrm{e}})(1 - C_{\mathrm{h}}) + (t_{\mathrm{b}} + t_x + t_{\mathrm{e}})C_{\mathrm{h}} \quad (2)$$

or

$$x_{\mathrm{co}} = t_{\mathrm{q}} + t_{\mathrm{h}} + t_{\mathrm{b}} + t_{\mathrm{e}} + t_x C_{\mathrm{h}} \quad (3)$$

And the expression for service time on *IntServ6* is,

$$x_{\mathrm{c}} = (t_i + t_{\mathrm{e}})(1 - C_{\mathrm{h}}) + (t_i + t_{\mathrm{e}} + t_x)C_{\mathrm{h}} \quad (4)$$

or

$$x_{\mathrm{c}} = t_i + t_{\mathrm{e}} + t_x C_{\mathrm{h}} \quad (5)$$

Equations (3) and (5) have a C_{h} parameter, which represents the collision rate. This parameter is introduced because of the use of a hash number to identify the flows on both technologies: *IntServ* and *IntServ6*. The collision rate is defined in References [11, 15] and is calculated as,

$$C_{\mathrm{h}} = 1 - \frac{N * \left(1 - \left(\frac{N-1}{N}\right)^m\right)}{m} \quad (6)$$

where N is the hash table size and m is the number of flows. With Equations (3), (5) and (6), and taking into account the geometric distributions mentioned above, we obtain the classification mean service time for each technology. Thus, the expression for *IntServ* mean service time is,

$$\bar{x}_{\mathrm{co}} = t_{\mathrm{q}} + t_{\mathrm{h}} + mt_{\mathrm{comph}} + t_{\mathrm{e}} + t_{\mathrm{comp}}mC_{\mathrm{h}}^{2} \qquad (7)$$

where the mean value of t_x was replaced by $m{*}Ch{*}t_{\mathrm{comph}}$; and t_{comph} is the time required for a comparison to evaluate a hash table row with the flow hash number. The mean value of t_{b} was replaced by $m{*}t_{\mathrm{comph}}$. This value was calculated based on a geometric distribution of the matching process time. Parameters t_{q}, t_{h} and t_{e} are constants and, for simulation proposes, their values have been calculated in Table 1 based on the number of *SRAM* or *SDRAM* readings needed for each task. Also, these calculations were performed taking into account the word processor size and taking the *IPv6 five-tuple* length as 296 bits. As an example of a typical network processor, we have selected the IXP 1200 Network processor. It was used for similar studies in References [3, 7, 8].

On the other hand, the mean service time for *IntServ6* is,

$$\bar{x}_{\mathrm{c}} = t_i + t_{\mathrm{e}} + t_{\mathrm{comp}}mC_{\mathrm{h}}^{2} \qquad (8)$$

where mean value of t_x was replaced by the same value explained above for Equation (7). Parameters t_i and t_{e} are constant and were calculated in Table 1 using the same principles mentioned above.

For simplicity, we suppose that packets arrive in bursts and, at the same time, such bursts arrive with a Poisson distribution. Also, we suppose that our router has a memory cache that holds each burst. As a result, it performs only one classification process by each packet burst. These assumptions are possible following the studies performed by Roberts in Reference [17], where it is clear that such packet behaviour can be obtained from typical Internet traffic based on TCP connections and applications such as the Web and FTP. Thus, we can model the classification process as an M/G/1 queue. We have considered a general distribution for the service discipline because there is a geometric distribution on the collision resolution process for both, IntServ and IntServ6, and also a geometric distribution in the hash table searching on IntServ. This is due to the fact that both processes perform a comparison with each table entry until a coincidence occurs.

However, as we want to use queuing networks principles to build our router model for future studies; we should obtain an M/M/1 model. Therefore, it is necessary to do some assumptions as follows. Normally, the classifier looks for a hash number on the collision table if a collision has occurred. Time search for this case has a geometric distribution. However, if more flow reservations are not created or dropped, the position on the Collision Resolution Table will be the same for the same flow. Thus, the classification service time will be the same for packets belonging to the same flow. But, if we consider that flow reservations are randomly created or dropped, we can assume that the flow location on the Collision Resolution Table of *IntServ* and *IntServ6* is randomly and exponentially distributed. We can make similar assumptions for the *hash table* of *IntServ*. As a result, we can assume that the service times for both processes (collision resolution for IntServ and *IntServ6*, and hash searching for *IntServ6*) have a Poisson distribution.

Then, under these assumptions, we can assume an M/M/1 tail for a specific session classification process. To perform this model, we can assume the same mean service time obtained for the M/G/1 model because the mean packet delay of the classification process is the same for all packets independently from the flow it belongs to.

7.2.3. *Packet forwarding model*

To obtain the router packet forwarding model, we take advantage of the M/M/1 model obtained for each particular process: classifier and scheduler. Then, we use the Kleinrock's independence principle (1976) and the Burke's Theorem to implement a tandem queues network. Therefore, the general expression for the mean E2E time (see Reference [18]) in this case is,

$$E(T) = \sum_{i=1}^{k} \frac{1}{\mu_i - \lambda_i} \qquad (9)$$

where λ_i and μ_i are the mean arrival rate and the mean service rate of each queue on the packets path, respectively.

Applying Equation (9) to our case, we can obtain the mean delay of the network composed by classifier and scheduler, both in tandem as:

$$\bar{d}_{\mathrm{r}} = \bar{d}_i + \bar{d}_{\mathrm{c}} \qquad (10)$$

where d_{r} is the router delay, d_i is the scheduling delay and d_{c} is the classification process delay. Parameter d_{c} is the average time that a packet spends in the classifier system, and it can be expressed as:

$$d_{\mathrm{c}} = \frac{1}{\mu - \lambda} \qquad (11)$$

or

$$d_c = \frac{1}{\frac{1}{\bar{x}_c} - \sum_{j=1}^{m} \lambda_j} \tag{12}$$

where Equation (12) is the same as Equation (11) but we have replaced the mean service rate (μ) by its equivalent expression in mean service time terms ($1/\bar{x}_c$). Also, we have replaced the total mean packets arrival rate (λ) by its equivalent expression as the sum of the mean arrival rates for all the ml input flows to the classifier. The mean packet delay for the classifier process is valid for any packet belonging to any of the m input flows.

However, for the scheduling process, we do not have an expression for the mean service time; but we have the expression (1) for the mean packet delay upper bound of the scheduler process instead. Thus, our model uses Equation (9) combining it with Equations (1) and (12), but the new expression obtained will have an upper bound for the router mean time delay:

$$\bar{d}_i + \bar{d}_c \leqslant \frac{\bar{L}_i}{w_i C - \frac{\bar{L}_i}{\bar{x}_c}} + \bar{d}_c \tag{13}$$

or

$$\bar{d}_i + \bar{d}_c \leqslant \frac{\bar{L}_i}{w_i C - \lambda_i \bar{L}_i} + \frac{1}{\frac{1}{\bar{x}_c} - \sum_{j=1}^{m} \lambda_j} \tag{14}$$

Then, we conclude that,

$$\bar{d}_r \leqslant \frac{\bar{L}_i}{w_i C - \lambda_i \bar{L}_i} + \frac{1}{\frac{1}{\bar{x}_c} - \sum_{j=1}^{m} \lambda_j} \tag{15}$$

where λ_1 is the router average packet arrivals rate. The structure of this expression is valid in general for an Integrated Services Router, but in this case, this expression can be used directly for *IntServ6* because we have used x_c as the service time parameter. If we wish to obtain the model specifically for the standard *IntServ*, then we should replace $\mu = \frac{1}{\bar{x}_{co}}$ and $\bar{x}_c = \bar{x}_{co}$, where $\bar{x}_{co}$ is given by Equation (7).

7.3. Mobility effect over the mean packet delay over Integrated Services wireless access routers

To evaluate the IntServ6 performance over mobile environments, we modelled a wireless access router. As the main QoS parameter for Integrated Services networks is the mean packet delay, we simulated this issue and its relationship with mobility parameters such as the new calls number and the handoff calls number. At the same time, these parameters depend on the admission control schemes.

To understand such relationship, we should see again the model introduced in Figure 8. This model can be applied for wireless access routers. It is more complete than others recently introduced such as Reference in [19].

Based on this model, we have developed a mathematical model which details several issues. The first modelled issue is the relationship between the number of flows (or calls) that the Flows reservations table holds and the new calls traffic intensity and handoff traffic intensity. In order to accomplish this, we modelled two admission control schemes, namely, the cuttoff priority scheme and the new call bounding scheme. The second modelled issue was the mean packet burst delay on the classifier. This parameter depends on the number of flows (or calls) that the reservations table contains. Next, we will explain both models and then we will show simulation results as follows.

7.3.1. Mobility model and admission control schemes

To model our approach, we have used the Fang Mobility Model described in Reference [20]. This model assumes that the arrival processes of new calls and handoff calls are Poisson. The channel holding time for new and handoff calls is assumed to be independent and exponentially distributed but with different average values. In Reference [20], Fang demonstrated that this is a simple and accurate approach.

Let λ_n, λ_h, $1/\mu_n$, $1/\mu_h$ denote the new and handoff call arrival rates, and the average channel holding times of new and handoff calls, respectively. Let $\rho_n = \lambda_n/\mu_n$, $\rho_h = \lambda_h/\mu_h$. Let C denote the total number of channels in a cell. Using the Fang mobility model, call level performance parameters, e.g. new call blocking probability (p_{nb}) and handoff dropping probability (p_{hb}), of QoS provisioning schemes for several CAC can be derived. The CAC schemes modelled in Reference [20] and used in our approach are explained below.

7.3.2. New call bounding scheme

In this scheme, we limit the admission of new calls into the wireless networks. The scheme works as follows: if the number of new calls in a cell exceeds a threshold when a new call arrives, the new call will be blocked; otherwise it will be admitted. A handoff call is rejected only when all channels in the cell are used up. The idea behind this scheme is that we would rather accept fewer customers than drop the ongoing calls in the future, because customers are more sensitive to call dropping than to call blocking. This scheme may work best when the call arrivals are bursty. When a big burst of calls arrives in a cell, if too many new calls are accepted, the network may not be able to handle the resulting handoff traffic, which will lead to a high call dropping. The new call bounding scheme, however, could handle the problem well by spreading the potential bursty calls (users will try again when the first few tries fail). On the other hand, as we observe in wired networks, network traffic tends to be self-similar. Wireless network traffic will behave the same considering more data services will be supported in the wireless networks. This scheme will be useful in the future wireless multimedia networks.

Let n_1 denote the number of new calls initiated in the cell and n_2 denote the number of handoff calls in the cell. Also, let K be the threshold for the new calls. Expressions for new call blocking probability ($p_{\text{nb-nc}}$) and handoff call probability ($p_{\text{hb-nc}}$) are given by,

$$p_{\text{nb}-\text{nc}} = \frac{\sum_{n_2=0}^{C-K} \frac{\rho_n^K}{K!} * \frac{\rho_h^{n_2}}{n_2!} + \sum_{n_1=0}^{K-1} \frac{\rho_n^{n_1}}{n_1!} * \frac{\rho_h^{C-n_1}}{(C-n_1)!}}{\sum_{n_1=0}^{K} \frac{\rho_n^{n_1}}{n_1!} \sum_{n_2=0}^{C-n_1} \frac{\rho_h^{n_2}}{n_2!}} \tag{16}$$

$$p_{\text{hb}-\text{nc}} = \frac{\sum_{n_1=0}^{K} \frac{\rho_n^{n_1}}{n_1!} * \frac{\rho_h^{C-n_1}}{(C-n_1)!}}{\sum_{n_1=0}^{K} \frac{\rho_n^{n_1}}{n_1!} \sum_{n_2=0}^{C-n_1} \frac{\rho_h^{n_2}}{n_2!}} \tag{17}$$

7.3.3. *Cutoff priority scheme*

In this scheme, instead of putting limitation on the number of new calls, we base on the number of total on-going calls in the cell to make a decision whether or not a new arriving call is accepted. The scheme works as follows.

Let m denote the threshold, upon a new call arrival. If the total number of busy channels is less than m, the new call is accepted; otherwise, the new call is blocked. The handoff calls are always accepted unless no channel is available upon their arrivals. This scheme has been studied in many papers [21–23], and analytical results for call blocking probabilities are obtained under the assumption that the average new call channel holding time and average handoff call channel holding time are equal so that one-dimensional Markov chain theory can be used. When the average channel holding times for new calls and handoff calls are different, the approach will not work. Then, the expressions for new calls blocking probability and handoff dropping probabilities are,

$$p_{\text{nb}-\text{cop}} = \frac{\sum_{j=m}^{C} \frac{(\rho_n+\rho_\text{h})^m \rho_\text{h}^{j-m}}{j!}}{\sum_{j=0}^{m} \frac{(\rho_n+\rho_\text{h})^j}{j!} + \sum_{j=m+1}^{C} \frac{(\rho_n+\rho_\text{h})^m \rho_\text{h}^{j-m}}{j!}} \tag{18}$$

$$p_{\text{hb}-\text{cop}} = \frac{\frac{(\rho_n+\rho_\text{h})^m \rho_\text{h}^{C-m}}{C!}}{\sum_{j=0}^{m} \frac{(\rho_n+\rho_\text{h})^j}{j!} + \sum_{j=m+1}^{C} \frac{(\rho_n+\rho_\text{h})^m \rho_\text{h}^{j-m}}{j!}} \tag{19}$$

7.3.4. *Number of flows at the wireless access router*

Now, we can use expressions Equations (16)–(19), given for p_{nb} and p_{hb} for both CAC models, to obtain the average number of new calls (N_n) and handoff calls (N_h) for each case. Therefore, we use the expression for the average number of elements over an Erlang Loss Queueing system given by Equation (20),

$$N = \frac{\lambda(1 - P_\text{b})}{\mu} \tag{20}$$

Then, we obtain N_n and N_h from p_{nb} and p_{hb}, respectively for each CAC model as,

$$N_n = \frac{\lambda_n(1 - P_{\text{nb}})}{\mu_n} \tag{21}$$

and

$$N_\text{h} = \frac{\lambda_\text{h}(1 - P_{\text{hb}})}{\mu_\text{h}} \tag{22}$$

Then, the mean total number of users on the cell for each CAC model is calculated as,

$$N_\text{t} = N_n + N_\text{h} \tag{23}$$

7.3.5. *Mean packet delay for the classification process*

This aspect was studied before on Subsection 7.2. Here, the mean packet delay for the classification process is given by Equation (12) and its m parameter (number of flows or calls) is given by Equation (23).

7.4. *Handoff latency analysis*

To analyse the handoff latency over an IntServ6 network, we start from the network architecture proposed in

Figure 5 and the messages interchange described in Figure 6. Although the message sequence is similar for both technologies, the main difference between the standard IntServ and IntServ6 is the way that tunnel reservations are managed. Thus, as we have described in Section 3 above, IntServ uses the RFC 2746 [9], whereas the IntServ6 proposal uses a different scheme described in that section. This aspect introduces some differences on the reservations setup time and the interchanged messages size.

First, RFC 2746 specifies that PATH messages sent to perform reservations setup, should carry a SESSION_ASSOC object; it has a length of 48 Bytes (it includes the common header, the SESSION object and the IPv6 FILTER_SPEC object). This object is useful to establish a relationship between the E2E reservations and the tunnel internal reservations.

On the other side, to establish reservations within tunnels, IntServ6 do not require to send any additional object, but change the IPv6 SENDER_TEMPLATE object (class = 11, c-type = 2), used for a simple reservation setup, for an IPv6 FLOW_LABEL SENDER_TEMPLATE object (class = 11, c-type = 3). This change is necessary to send the hash number that identifies the reservation. As both objects have the same length (20 Bytes), the transmitted Bytes number remains unchanged respect to a simple reservation case, this is, a reservation that do not cross a tunnel.

In conclusion, every time a new reservation over a tunnel is established into the handoff process, IntServ6 will save the transmission time necessary to send 48 Bytes of the SESSION_ASSOC object in IntServ.

Additionally, in the new reservation setup, IntServ6 saves processing time necessary to calculate the hash number on every router over the new reservation path (including the tunnel routers). This process is not performed due to the hash number is transmitted previously in the PATH message, as described in Sections 2 and 3.

As a result, the saved time (T_s) on the IntServ6 handoff process is given by

$$T_s = \Sigma\left(\left(\frac{1}{V_{L-i}}\right) * 48 * 8\right) + \Sigma(T_{chash-j}) \qquad (24)$$

where $V_{L\text{-}i}$ is the ith link capacity (in bits/s) in the new path and $T_{chash-j}$ is the hash calculation time for the jth router in the new path.

The hash calculation time can be obtained as the number of readings and writings necessary to read the five-tuple and to write the hash key. If we use the same IXP1200 network processor as an example, a total of 10 SRAM readings and 1 SRAM writing are necessary to read 296 bits belong to the five-tuple and 20 bits belong to the hash number. Thus, the hash number calculation time is approximately 880 ns.

Finally, if we use the example network in Figure 5, and if we suppose that the wireless link is working with a capacity of 11 Mbps and the fixed link between the MAP and the Access router is working with a capacity of 100 Mbps, we obtain that for a local handoff, the IntServ6 saved time is 40.5 μs, whereas in a Regional handoff (between different MAPs), the IntServ6 saved time in the handoff process is 41.38 μs.

8. SIMULATION RESULTS

The models described in Section 7 were simulated and important results were obtained. In this section, results for fixed and mobile routers, with both QoS technologies (IntServ and IntServ6), will be described.

8.1. Fixed routers results

First, we developed simulations for fixed routers. Then, we only take into account the packet forwarding model without introducing the mobility effect. Thus, we take the expressions given in Subsection 7.2. Figure 10 shows the mean burst packet delay for both QoS technologies. As the classification process is the main difference between standard IntServ and IntServ6, then, we have performed simulations for the classification process to evaluate our approach over *IntServ* and *IntServ6* technologies. Here, we assumed that the scheduler model had already been tested as in Reference [16] and thus, we do not need to test it anymore. Results for total router delay can be obtained easily by adding the scheduler delay bound given by Equation (1) to the classifier delay given by Equation (12).

Simulations were developed based on an IXP1200 network processor (other similar studies using this network processor are described in References [3, 7, 8]) with parameters given by Table 1 and for a hash table size of 2^{20}. This value was assumed to compare both technologies because it is the *IntServ6* hash table size. We consider 128 bursts/s of mean burst arrival rate on each flow. The mean delay time for these simulations was obtained for 50 bursts.

We can observe in Figure 10 that in both QoS technologies (*IntServ* and *IntServ6*), if the number of flows increases then the mean classification delay will increase as well. This is because the collisions number has

DOI:10.1002/ett

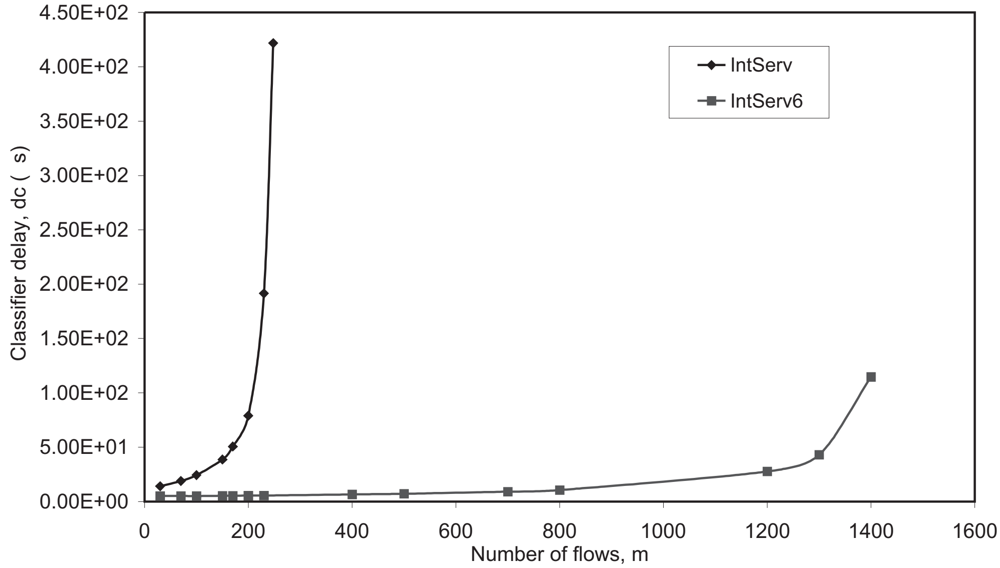


Figure 10. Mean classification delay comparison between technologies.

increased. However, if we compare *IntServ* and *IntServ6*, we observe that the mean classification delay for *IntServ6* is less than *IntServ* in several magnitude orders. This is due to the tagging scheme for classification on *IntServ6*. It is faster than the matching classification process done by *IntServ*. Also, *IntServ6* has a better performance with respect to *IntServ*, because the hash number calculations are done by the End system and not by the router with each packet arrival as *IntServ* does. In Figure 10, *IntServ* simulation was performed only up to 248 flows. For more flows the mean service rate is less than the mean arrival rate and the queue length will increase drastically. Thus, our theoretical mean delay given by Equation (12) cannot be applied. With *IntServ6* this problem is reduced as we can see in the same Figure 10. Then, an IntServ6 router can process 1200 flows without great performance looses.

8.2. Wireless access routers results

Finally, we have simulated a wireless access router. Here, we have considered the mobility effect over the mean packet delay, which has been introduced in Subsection 7.3. Besides, we used the issues of an IXP1200 network processor (see Table 1) to obtain the mean service time of the classification process. We obtained the mean classification delay upper bound dependence with respect to the traffic intensity variation, which was represented in the CAC models as $\rho = \lambda/\mu$. Also, here we do not show the results for the upper delay bound on the router, but we do show the classifier delay results because it is the main advantage of our approach. Then, we simulate several scenarios: first, a scenario with new calls rate variations, and a second scenario with handoff calls variations. Such scenarios were simulated for both QoS technologies: IntServ6 and standard IntServ, and for two CAC schemes: the new call bounding scheme and the cutoff priority scheme. All simulations were performed with $C = 30$ resources to assign on the access router. Each resource has a mean burst arrival rate of 128 burst/s. To perform our simulations we select ρ_{n} and ρ_{h} values so that N_{t} should be an Integer value because in practice, the number of calls is an integer value. Also, we take into account that the number of allocated resources cannot be more than the higher value obtained with each CAC scheme. This is, each CAC can only assign as much as a maximum number of resources, thus N_{t} reaches an upper bound under congestion conditions (27 resources for the new call bounding scheme and 25 resources for the cutoff priority scheme).

First, in Figure 11 we simulate the mean classification delay variation with the new calls traffic intensity variation for both CAC schemes and both QoS technologies (IntServ and IntServ6). Here, the handoff traffic intensity is constant and equal to 15. Secondly, in Figure 12 we performed simulations for the classifier delay versus handoff calls

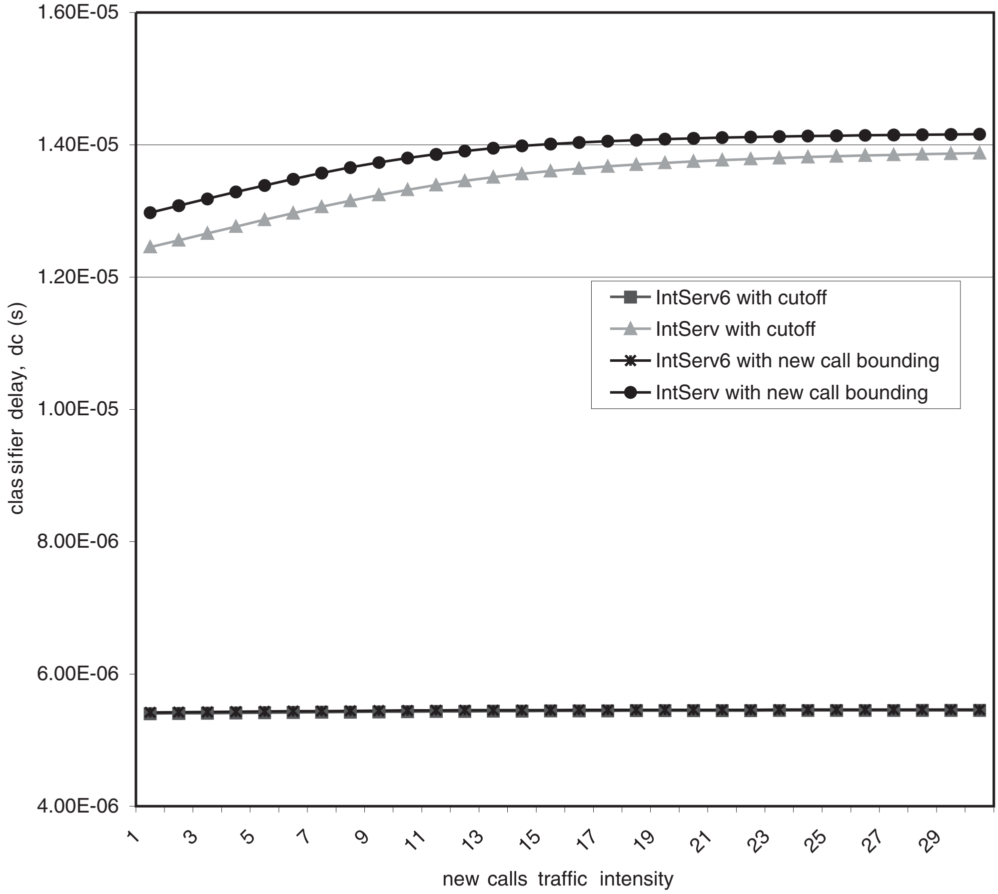


Figure 11. Mean classification delay for IntServ and IntServ6 under two CAC schemes versus new calls traffic intensity.

traffic intensity variations for both CAC schemes and both QoS technologies. In this case, the new calls traffic load is constant and equal to 15.

We can observe in Figures 11 and 12 that the mean classification delay for the standard IntServ increases when the number of flows increases. On the other hand, the IntServ6 mean classifier delay increases slowly. Thus, for IntServ6, the mean packet delay variation with the number of calls is almost zero. Besides, our approach has a classification delay smaller than IntServ delay, i.e. the IntServ6 Mean Classification Time is less than one magnitude order respect to the same parameter for standard IntServ.

Also, in Figures 11 and 12 we can see that several CAC schemes have different effect over the mean classifier delay for both QoS technologies. Thus, the cutoff priority scheme presents a classifier delay smaller than the new call bounding scheme. However, in both CAC schemes, we can observe that the mean classifier delay for IntServ routers reaches an upper bound. This effect occurs because the CAC scheme reaches the maximum number of resources it can assign. Thus, when the wireless access router is working with all its resources assigned, under these conditions, the mean number of flows remains constant and then, the mean number of searches in the classification process is constant. On the other hand, for IntServ6 this effect is less important because the number of searches in the hash table is only one and does not depend on the number of flows. Then, for IntServ6 the number of searches has only small variations that appear because of Hash collisions. Thus, the mean classification delay behaviour respect to the number of flows is almost flat for both CAC schemes; then, both IntServ6 curves look like only one curve in Figures 11 and 12. In conclusion, IntServ6 mean classification delay has a very low dependence with the user mobility; whereas IntServ mean classification delay has a high dependence with the user mobility.

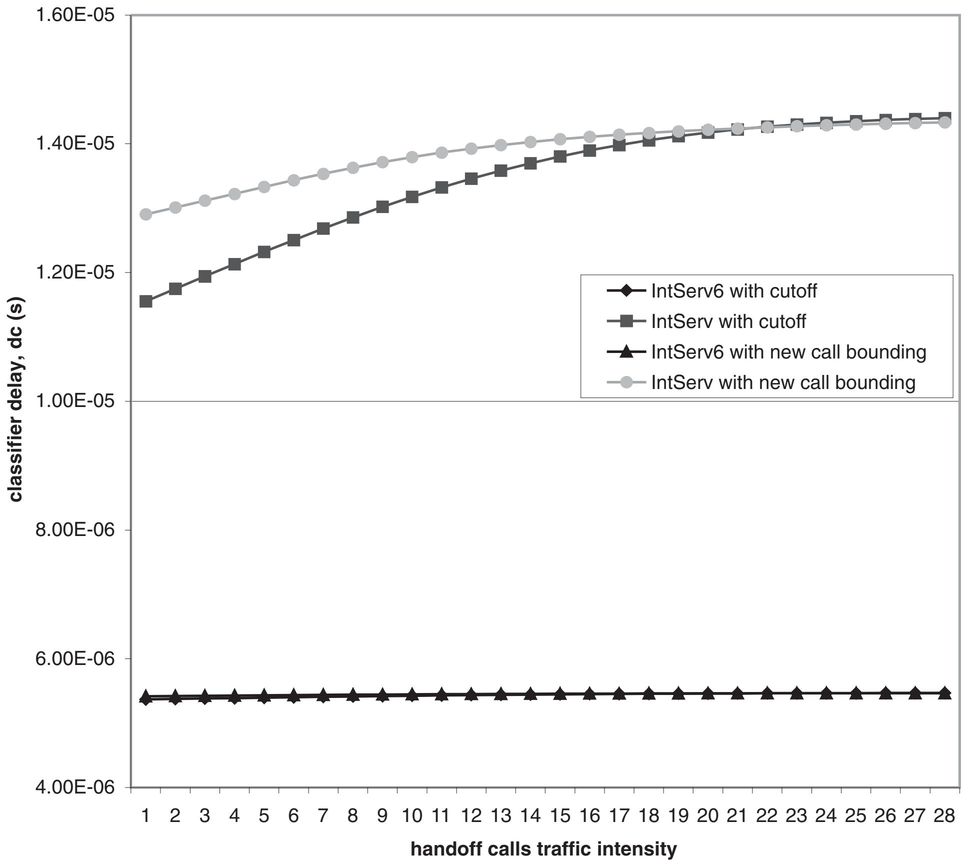


Figure 12. Mean classification delay for IntServ and IntServ6 under two CAC schemes versus handoff calls traffic intensity.

9. CONCLUSIONS

In this paper we have described a new approach to support QoS on IPv6 networks. This approach was named IntServ6 and it is based on ISA. IntServ6 presents a better router performance and improves other issues of the standard IntServ approach. A first advantage is that IntServ6 improves the router performance with respect to IntServ. Due to the reduction of the mean classification delay, the number of flows that an IntServ6 router can serve is higher than the same parameter in an IntServ router. Other important issues improved by IntServ6 are the reservations within tunnels, the flows aggregation and the interconnection with MPLS transport networks. With IntServ6, the reservations within tunnels problem is solved at the network layer whereas IntServ solves the problem at the transport layer. On the other hand, IntServ6 resolves the problem of flows aggregation in a natural way by means of tunnelling principle, which is similarly used as the MPLS label stacking procedure. On the contrary, the flows aggregation problem on IntServ is resolved by means of other technologies such as DiffServ and not by the same IntServ. Besides, the flows aggregation issue on IntServ6 reduces the scalability problem that IntServ has nowadays. Finally, IntServ6 brings forth a more efficient way for interconnection methods to MPLS networks.

In addition, we have described some IntServ6 extensions for wireless networks. To reduce the handoff latency our network architecture should support the user mobility by means of the HMIPv6 protocol. Also, to reduce the impact of changes over HMIPv6, our approach uses RMA. To evaluate our approach under mobile environments at a QoS level, we simulated the mobility effect over the mean classification delay. Our simulations results show that several CAC schemes have different effect over the classification delay on the wireless access routers. Thus,

DOI:10.1002/ett

the cutoff priority CAC Scheme has a smaller delay than the new call bounding CAC Scheme. Finally, the mean classification delay on an IntServ6 based wireless access router is less affected by the mobility parameters that on an IntServ based wireless access router.

ACKNOWLEDGEMENTS

This work was supported by ERDF and the Spanish government through projects TIC2003-01748 and TIC2006-04504.

REFERENCES

1. Deering S. *Internet Protocol version 6 (IPV6). IETF RFC* 2460, 1998, pp. 1–39.
2. Conta A, Deering S. *IPv6 flow label Specification. IETF RFC* 3697, 2004, pp. 1–9.
3. Padilla J, Huerta M, Paradells J, Hesselbach X. IntServ6: an approach to support QoS over IPv6 Networks. *10th IEEE Symposium on Computers and Communications Proceedings*, 2005, pp. 77–82.
4. Padilla J, Paradells J, Rodríguez A. Effects of mobility on mean packet delay over integrated services wireless networks. *2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06)*, 2006.
5. Braden R. Integrated services in the internet architecture: an overview. *RFC* 1633, 1994.
6. Wang Z. *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann, San Francisco, USA, 2001; p. 256.
7. Kaur J. Scalable network architectures for providing per-flow service guarantees. *Dissertation*, The University of Texas at Austin: Austin, Texas, 2002, p. 153.
8. Hardekopf B, Riché T, Mudigonda J, Dahlin M, Vin H. Impact of network protocols on programmable router architectures. *Laboratory of Advanced Systems Research, Department of Computer Science, University of Texas at Austin* 2003, p. 12.
9. Terzis A. RSVP Operation Over IP Tunnels. *RFC* 2746, 2000, pp. 1–25.
10. Baker F, Iturralde C. *Aggregation of RSVP for IPv4 and IPv6 Reservations. IETF RFC* 3175, 2001, p. 1–36.
11. Padilla J, Paradells J, Rodríguez A. Supporting QoS over IPv6 wireless networks with IntServ6. *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, 2006.
12. Soliman H, Castelluccia C. *Hierarchical Mobile IPv6 mobility Management (HMIPv6). IETF RFC 4140*, 2005, p. 1–29.
13. Koodli R. *Fast Handovers for Mobile Ipv6. IETF RFC* 4068: 2005, p. 1–42.
14. Paskalis S, Kaloxylos A, Zervas E, Merakos L. 2003. An efficient RSVP-mobile IP interworking scheme. *Mobile Networks and Applications* 2003; **3**:197–207.
15. Wang Z. *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann, San Francisco, USA, 15 March 2001, p. 256.
16. Zhang J, Hamalainen T, Joutsensalo J. stochastic analysis of upper delay bound of GPS-based packetized fair queueing algorithms. *ICON*, 2004.
17. Bonald T, Roberts JW. Congestion at flow level and the impact of user behaviour. *Computer Networks* 2003; **42**:521–536.
18. Chee Hock N. *Queueing Modeling Fundamentals*. John Wiley & Sons Ltd., Chichester, UK, 1997; p. 222.
19. Yu F, Ma L, Leung V. Effects of mobility on packet level performance of connection-oriented multimedia traffic over packet-switched cellular networks. *IEEE Wireless Communication & Networking Conference* 2004; **4**:2171–2176.
20. Fang Y, Zhang Y. Call admission control schemes and performance analysis in wireless mobile networks. *IEEE Transactions on Vehicular Technology* 2002; **2**:371–382.
21. Hong D, Rappaport SS. Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoff procedures. *IEEE Transactions on Vehicular Technology* 2000; **5**:2037–2039.
22. Li B, Lin C, Chanson S. Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks. *Wireless Networks* 1998; **4**:279–290.
23. Yoon CH, Un CK. Performance of personal portable radio telephone systems with and without guard channels. *IEEE Selected Areas on Communications* 1993; **6**:911–917.

AUTHORS' BIOGRAPHIES

Jhon Padilla received his M.Sc. degree in Computer Science in 1998 from Universidad Industrial de Santander in Colombia. Currently, he is a Ph.D. candidate from Universidad Politécnica de Cataluña in Spain. He is a Professor at the Electronic Engineering Faculty at Universidad Pontificia Bolivariana in Colombia. His primary research interests include wireless networks, next generation networks and Quality of Service.

Josep Paradells received his Ph.D. Degree in Telecommunications Engineering at Universidad Politécnica de Cataluña (UPC), Spain. Currently, he is a Professor in the Telematic Engineering Department at the same UPC. His research areas are wireless networks, traffic analysis and data networks modelling.