Supporting QoS over IPv6 wireless networks with IntServ6

Jhon J. Padilla
Pontificia Bolivariana University
Bucaramanga, Colombia

Josep Paradells
Technical University of Catalonia
Barcelona, Spain

Alvaro Rodríguez
Pontificia Bolivariana University
Bucaramanga, Colombia

ABSTRACT

In this paper we introduce a new approach to support QoS on wireless networks. It uses a new solution for QoS support on IPv6 networks named IntServ6 and introduce some extensions to bring resource reservations mobility. Our solution works over Hierarchical Mobile IPv6 networks. IntServ6 takes advantage of the IPv6 header flow label field to improve the router performance. It also improves the current standard solution for the resource reservation within tunnels for integrated services networks. Evaluation results show that this approach has a better router performance respect to the traditional Integrated Services solutions for mobile networks. [1]

I. INTRODUCTION

Next generation networks will be based on IPv6 protocol. As a result, these networks will have issues that the standard IPv4 can not support such as a great IP addresses number, users mobility, quality of service, security, etc. Such networks will have to support several services classes for elastic applications and real time applications and will have to support the user mobility.

Our approach, named IntServ6, is based on the Integrated Services Architecture but with an important difference in the classification process: while standard IntServ uses the five-tuple for the packet classification, IntServ6 uses the IPv6 flow label field to identify the flow that 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 utilization has not been completely standardized. Although current Standards [1] describe general recommendations, 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. Also, this IntServ6 issue allows a simple solution to the "Reservations within Tunnels" problem over current Hierarchical mobile IP based networks. On the other hand, for solving this problem, the current IntServ standard has a very complicated solution.

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 on [2]. It describes a router performance model with parameters such as the required number of threads and the memory bandwidth. A second model, detailed here, describes the mean packet delay on a router. To explain our approach, this paper is organized as follows: section II describes the classification process for the Standard IntServ and IntServ6. In section III, we explain the reservations within tunnels problem and the solutions for IntServ and IntServ6. Section IV describes our network architecture and the reservations mobility problem; also, this section introduces the IntServ6 extensions to resolve this problem. Section V presents our mathematical models and section VI, shows our simulation results. Finally, section VII shows our conclusions.

II. RESERVATIONS SUPPORT WITH THE FLOW LABEL FIELD

IntServ6 is based on the IntServ architecture for QoS support. In order to improve the 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, the packet classification process is executed 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 Fig. 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 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 [2-4]). The use of the Flow label field was described in [1, 5], 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 independently from 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 Fig. 1), which holds information about collided flows.

[1] This work was supported by ERDF and the Spanish government through project TIC2003-01748.

1-4244-0330-8/06/$20.00©2006 IEEE

Resource Reservation Table

Reservation State Id.	Output link	Source	Input Link	Previous hop	Reserved Resources	Refresh time	Requirement State
7	L2	H1	L1	H1	B	1000	Solicited
8							
9	L2	H4	L6	R2	B	500	Reserved

Hash Table

Flow Label	collision	Reservation State Id.
5000	No	7
5025	Yes	Null

Collision Resolution Table

Flow Label	Five-tuple	Reservation State Id.
5025	O1,D1,sp1,dp1, pID1	9
5025	02,D2,sp2,dp2, pID2	8

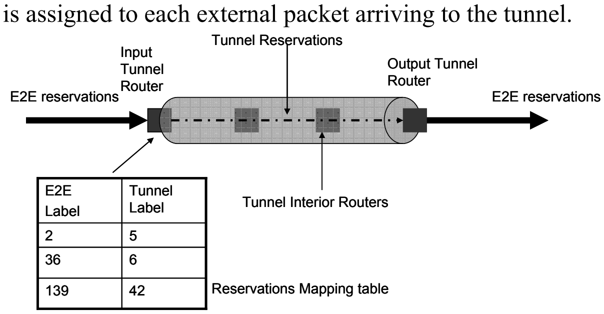
Figure 1. Reservation table management in IntServ6

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.

III. RESERVATIONS WITHIN TUNNELS

One important aspect to resolve in IP tunneling 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 can not be read from the external IP header. The solution for Standard IntServ is described on [6]. This solution resolves the problem at the transport layer. Thus, to distinguish between several reservations within the tunnel, in that approach [6] is necessary to transport the external IP packets within UDP tunnel packets. As a result, one UDP port for each resource reservation is assigned.

On the other hand, our approach gives a simple solution to the hidden Five-tuple problem within the tunnels. In IntServ6 (see Fig. 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.



E2E Label	Tunnel Label
2	5
36	6
139	42

Figure 2. Reservations within tunnels with IntServ6.

This new label is named Tunnel Label. It is carried on the external IPv6 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 on this document.

In order to achieve a better process understanding, first we should define two reservation types: The End to End (E2E) Reservations, which are resource reservations allocated before and after the tunnel; and the 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 mean 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 the 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. Also, 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. Then, 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 recognized by the Tunnel Interior routers, meanwhile PATH and RESV messages from the Tunnel Reservation travel without the encapsulating process and they are identified by the Tunnel Interior routers.

IV. 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. The main problem for solving 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 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 for establish the new reservation and to delete the old reservation. To reduce the handoff latency, IntServ6 could operate under a Hierarchical Mobile IPv6 (HMIPv6) [7] network. With the use of HMIPv6, the changed part of the

1-4244-0330-8/06/$20.00©2006 IEEE

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 even more the handoff latency is the Fast Handoff protocol [8]. As a result, our network architecture (see fig. 3) uses HMIPv6 for mobility support. Another important issue of our solution is that it does minimum modifications to the standard mobility protocols. Thus, to support the reservations mobility, our approach is based on a solution proposed by Paskalis [9] but with some modifications. Due to paper space requirements this solution does not be explained here, only we will explain our approach.

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 a 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 Access Routers (AR) Level but without MAP changes. A second situation is the Inter-domain handoff; this is, a handoff could occur between ARs belonging to different MAPs. The main situations are described in fig. 4, 5(a) and 5(b), where a local handoff and inter-domain handoff are described respectively. In order to give a simple explanation, we only describe a local handoff with the Correspondent Node as data source, and the Mobile Node as the data destination.

As we can see on fig. 4, 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 acknowledge 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 sets up the new reservation until the MN. Once the new reservation is established, the old reservation is dropped by mean of PATH Tear and RESV Tear messages between the RMA and the old Access Router (oAR). The tunnel reservation set up is not showed on these figures because we assume this process is implicitly performed when the tunnel is established between the MAP and the MN.

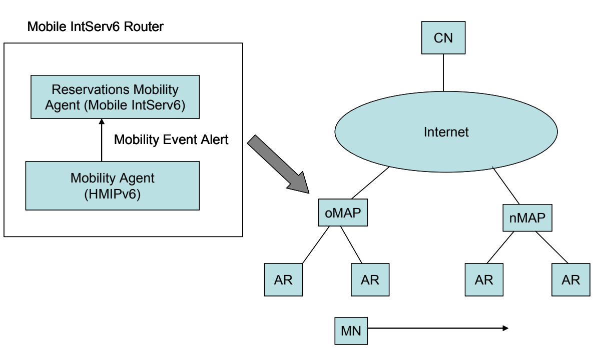


Figure 3. Mobile IntServ6 network architecture

V. INTSERV6 EVALUATION OVER MOBILE ENVIRONMENTS

To evaluate the IntServ6 performance over mobile environments, we modeled 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 have introduced in Fig. 6 a wireless access router model. We can observe that an Integrated Services wireless Access Router is mainly composed by two parts, first part is the reservations manager, which is composed by the admission control scheme and the reservations setup agent.

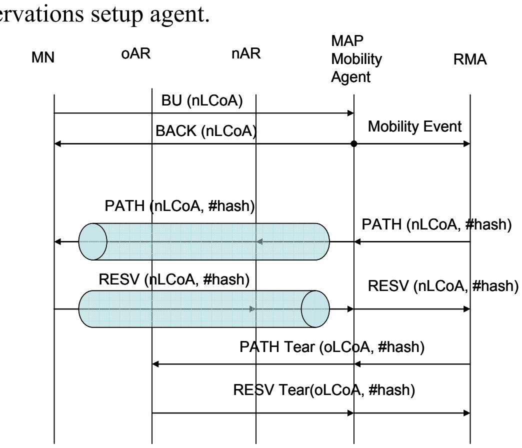


Figure 4. Reservations Local Handoff process on Mobile IntServ6.

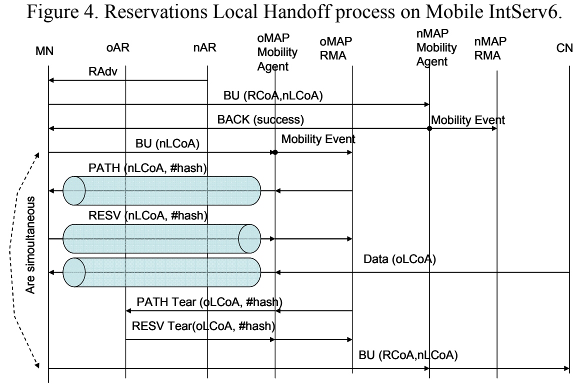


Figure 5(a). Reservations Inter-domain Handoff process on Mobile IntServ6

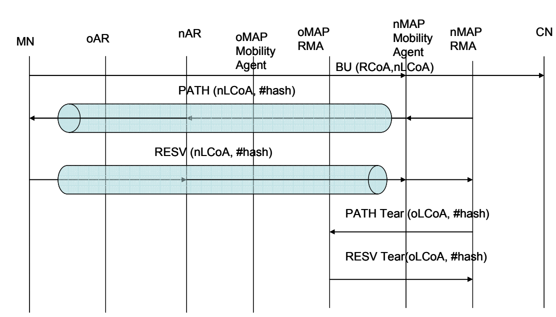


Figure 5(b). Reservations Inter-domain Handoff process on Mobile IntServ6 (continuation).

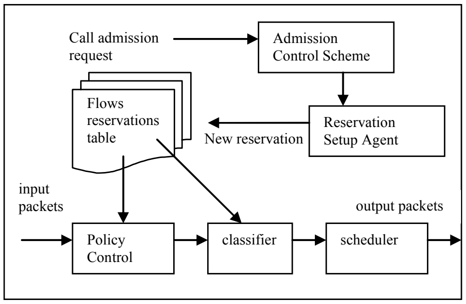


Figure 6. Wireless access router model.

The second part performs the packet forwarding process. It is composed by the policies control module, the classification process and the scheduling process. As we can see in fig. 6, 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, in other words, it should be found on the Flows reservations table. Finally, it should be scheduled over the correspondent output link.

Based on the described model, we have developed a mathematical model which details several issues. The first 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 modeled two admission control schemes: the Cuttoff Priority scheme and the New Call bounding scheme. The second modeled 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.

A. *Mobility model and admission control schemes*

To model our approach, we have used the Fang Mobility Model described on [10]. This model assumes that the arrival processes of new call and handoff calls are Poisson. The channel holding time for new and handoff calls is assumed independent and exponentially distributed but with different average values.

Let λ_n, λ_h, $1/\mu_n$ y $1/\mu_h$ denote the new and handoff call arrival rates, and the average channel holding times of new and handoff calls, respectively. Let $\rho_n=\lambda_n/\mu_n$ y $\rho_h=\lambda_h/\mu_h$. Let C denote the total number of channels in a cell. Using the Fang mobility model, call level performance, 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 modeled in [10] and used in our approach are explained below.

1) *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.

Let n_1 denotes the number of new calls initiated in the cell and n_2 denotes 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_{nb-nc}) and handoff call probability (P_{nh-nc}) are given by,

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

$$p_{hb-nc}=\frac{\sum_{n_1=0}^{K}\frac{\rho_n^{n_1}}{n_1!}*\frac{\rho_h^{C-n_1}}{(C-n_1)!}}{\sum_{n_1=0}^{K}\frac{\rho_n^{n_1}}{n_1!}\sum_{n_2=0}^{C-n_1}\frac{\rho_h^{n_2}}{n_2!}} \quad (2)$$

2) *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 a new arriving call is or is not 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 the arrivals. Then, the expressions for new calls blocking probability and handoff dropping probabilities are,

$$p_{nb-cop}=\frac{\sum_{j=m}^{C}\frac{(\rho_n+\rho_h)^m\rho_h^{j-m}}{j!}}{\sum_{j=0}^{m}\frac{(\rho_n+\rho_h)^j}{j!}+\sum_{j=m+1}^{C}\frac{(\rho_n+\rho_h)^m\rho_h^{j-m}}{j!}} \quad (3)$$

$$p_{hb-cop}=\frac{\frac{(\rho_n+\rho_h)^m\rho_h^{C-m}}{C!}}{\sum_{j=0}^{m}\frac{(\rho_n+\rho_h)^j}{j!}+\sum_{j=m+1}^{C}\frac{(\rho_n+\rho_h)^m\rho_h^{j-m}}{j!}} \quad (4)$$

3) *Number of flows at the wireless Access Router*

Now, we can use expressions (1) to (4), 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 (5),

$$N=\frac{\lambda(1-P_b)}{\mu} \quad (5)$$

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

$$N_n=\frac{\lambda_n(1-P_{nb})}{\mu_n} \quad (6)$$

and,

$$N_h=\frac{\lambda_h(1-P_{hb})}{\mu_h} \quad (7)$$

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

$$N_t=N_n+N_h \quad (8)$$

1-4244-0330-8/06/$20.00©2006 IEEE

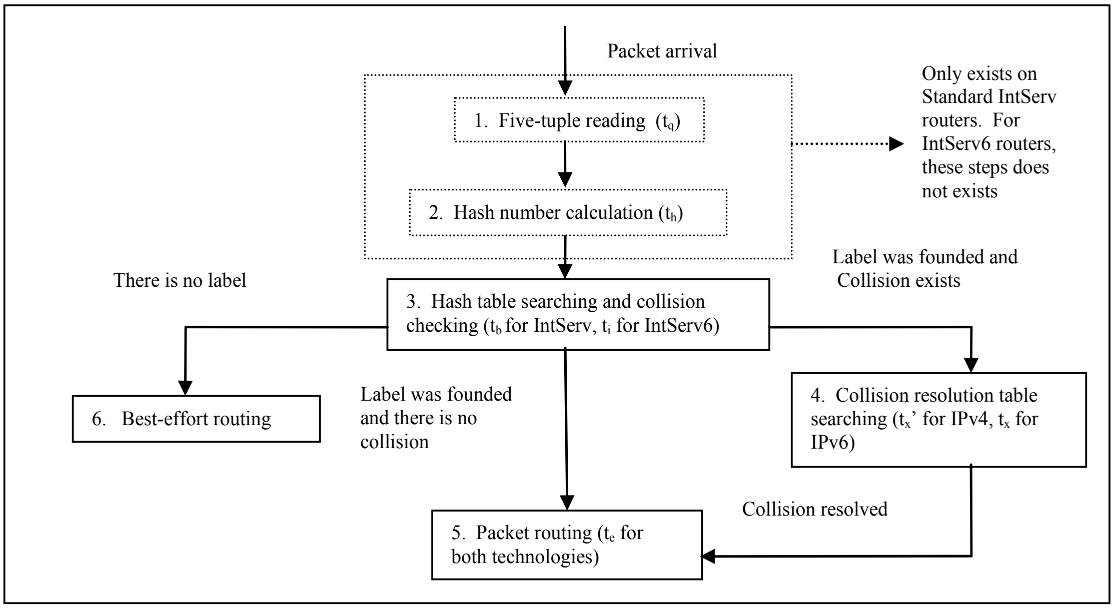


Figure 7. State transition diagram for the classification process on Integrated Services

B. Mean Packet Delay for the classification process

We have obtained a stochastic model for the classifier mean packet delay, which is based on the states transition diagram (STD) of the process (see Fig. 7). This STD changes for each technology (IntServ and IntServ6); thus, the obtained mathematical expressions for each one also change. Main differences between IntServ and IntServ6 are that in IntServ6 the five-tuple reading and its calculation are not performed at the router; meanwhile in IntServ these actions are required. In addition, the hash table searching in IntServ6 uses a tagging scheme, meanwhile this process in IntServ uses a matching scheme (see [3,4]). Time parameters for the obtained expressions are: t_q (five-tuple reading time), t_h (hash number calculation time), t_{comp} (five-tuple comparison time), t_{comph} (comparison between hash number and a hash table row on IntServ), t_e (packet routing time) and t_i (hash table verification time).

Thus, the obtained expression for the mean service time on IntServ is,

$$\bar{x}_{co} = t_q + t_h + mt_{comph} + t_e + t_{comp} mC_h^2 \tag{9}$$

And the expression for service time on IntServ6 is,

$$\bar{x}_c = t_i + t_e + t_{comp} mC_h^2 \tag{10}$$

Equations (9) and (10) have a C_h parameter, which represents the collision rate. This parameter is introduced due to the use of a hash number to identify the flows on both technologies: IntServ and IntServ6. The collision rate is defined in [11] and is calculated as,

$$C_h = 1 - \frac{N * \left(1 - \left(\frac{N-1}{N}\right)^m\right)}{m} \tag{11}$$

Where N is the hash table size and m is the number of flows.

For simplicity, we suppose that packets arrive in bursts and, at the same time, these burst arrive with a Poisson distribution. Also, we suppose that our router has a memory cache that holds each burst and allows performing only one classification process by each packet burst. Thus, we can assume that the arrivals to the classification process have a Poisson distribution. Then, we can model the classification process as an M/G/1 tail. 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.

However, as we want to use queuing networks principles to build our router model in future studies; we should obtain an M/M/1 model. Therefore, it is necessary to do some assumptions. 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. But, if we consider that flow reservations are randomly created or dropped, we can assume that the flow allocation 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. Thus, we can assume that the classifier service time 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. As a result, the mean packet classification delay (d_c) is the average time that a packet spends in the classifier system, and it can be expressed as:

$$d_c = \frac{1}{\mu - \lambda} \tag{12}$$

Or,

$$d_c = \frac{1}{\frac{1}{\bar{x}_c} - \sum_{j=1}^{m} \lambda_j} \tag{13}$$

VI. SIMULATION RESULTS

Fig. 8 and 9 show the simulation results for the mean packet delay behavior for the classification process respect to the new calls and handoff calls traffic intensity for both CAC schemes mentioned before. We show the mean classification delay because it represents the main difference between IntServ and IntServ6 routers. Our simulations are based on an IXP 1200 network processor hardware parameters and we suppose that the maximum number of calls in a cell is 30 and all the flows (or calls) require the same number of resources. Also, for the Fig. 8 results, we have supposed that the handoff calls traffic intensity is constant and equal to 15. At the same time, for results showed on Fig. 9, we have supposed that the new calls traffic intensity is also constant and equal to 15. In both figures, results show that the mean classification delay is affected by the CAC scheme used on the wireless access router. Also, we can observe that the mean classification time for IntServ6 is lower than the standard IntServ solution. Finally, we can observe that the mean packet delay on an IntServ6 based wireless access router is less affected by the mobility parameters that an IntServ based wireless access router.

VII. CONCLUSIONS

We proposed an approach to support QoS over mobile networks. Our approach reduces the packet classification delay and thus, it speeds up the router performance. IntServ6 has a more efficient solution for the "Reservations within tunnels" problem respect to the standard IntServ approach. This aspect is important because our solution is based on the Hierarchical Mobile IPv6 standard to support the user mobility. This approach requires tunneling between the Mobility Anchor Point and the Mobile host.

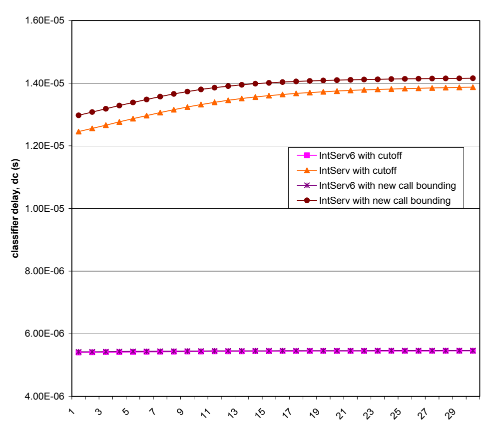


Figure 8. Mean packet delay for IntServ and IntServ6 under two CAC schemes vs. new calls traffic intensity.

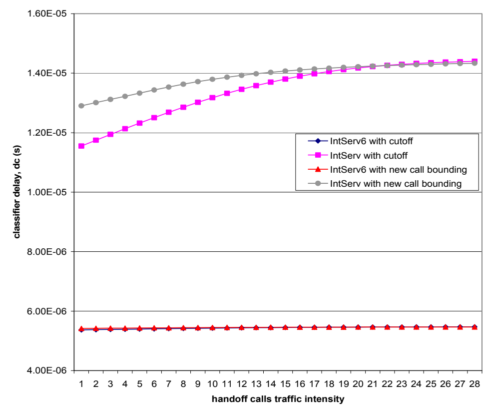


Figure 9. Mean packet delay for IntServ and IntServ6 under two CAC schemes vs. handoff traffic intensity.

On the other hand, simulation results show that the mean classification delay is affected by the CAC scheme used on the wireless access router. Also, we can observe that the mean classification time for IntServ6 is lower than the standard IntServ solution. Finally, we can see that the mean packet delay on an IntServ6 based wireless access router is less affected by the mobility parameters that an IntServ based wireless access router.

REFERENCES

[1] A. Conta and S. Deering. IPv6 Flow Label Specification. *IETF RFC 3697*, pp. 1-9, 2004.

[2] J. Padilla, M. Huerta, J. Paradells, and X. Hesselbach. IntServ6: An Approach to Support QoS over IPv6 Networks. *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, pp. 77-82, 2005.

[3] J. Kaur. *Scalable Network architectures for providing per-flow service guarantees*. Austin, Texas: Dissertation. The University of Texas at Austin, 2002.

[4] B. Hardekopf, T. Riché, J. Mudigonda, M. Dahlin, and H. Vin. *Impact of Network Protocols on Programmable Router Architectures*. Austin, Texas: Laboratory of Advanced systems Research, Department of Computer Science, University of Texas at Austin, 2003.

[5] S. Deering. Internet Protocol version 6 (IPV6). *IETF RFC 2460*, pp. 1-39, 1998.

[6] A. Terzis. RSVP Operation Over IP Tunnels. *IETF RFC 2746*, pp. 1-25, 2000.

[7] H. Soliman and C. Castelluccia. Hierarchical Mobile IPv6 mobility Management (HMIPv6). *IETF RFC 4140*, pp. 1-29, 2005.

[8] R. Koodli. Fast Handovers for Mobile Ipv6. *IETF RFC 4068*, pp. 1-42, 2005.

[9] S. Paskalis, A. Kaloxylos, E. Zervas, and L. Merakos. An Efficient RSVP-Mobile IP Interworking Scheme. *Mobile Networks and Applications*, vol. 8, pp. 197-207, 2003.

[10] Y. Fang and Y. Zhang. Call admission control schemes and performance analysis in wireless mobile networks. *Vehicular Technology, IEEE Transactions on*, vol. 51, pp. 371-382, 2002.

[11] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, 1st edition ed: Morgan Kaufmann, March 15, 2001.

1-4244-0330-8/06/$20.00©2006 IEEE