Improvement in Resource Reservations Management within Tunnels over All-IP Mobile Networks using IntServ6

Padilla J.
Electronic Engineering Department
Pontificia Bolivariana University

Paradells J.
Telematic Department
Technical University of Catalonia

Abstract- **Next Generation Mobile Networks (NGN) will be based on IPv6. Then, NGNs will require Mobility Protocols such as MIPv6 and HMIPv6. Also, they should support Quality of Service by means of technologies such as the Integrated Services architecture. In this context, a problem to distinguish reservations arises when Resource Reservations within Tunnels are required. The Current solution to this problem in the standard IntServ has some disadvantages such as reservations differentiation at the Transport layer level. However, a new approach to support QoS over Integrated Services Mobile Networks has appeared. It is named IntServ6. This approach resolves the Reservations within Tunnels problem in a natural and simple way by means of the Flow Label field of the IPv6 packet header. Subsequently, it resolves this problem at the Network Layer level. This paper describes and evaluates such approach to support Reservations within Tunnels. Both QoS technologies, IntServ and IntServ6 are compared. Results show that IntServ6 serves more traffic than IntServ over Simple Reservations and Reservations within tunnels.***

I. INTRODUCTION

Next Generation Mobile Networks (NGN) will be based on IPv6, which will require Mobility Protocols such as MIPv6 [1] (Mobile IP for IPv6) and HMIPv6 [2](Hierarchical Mobile IP for IPv6). Furthermore, they should support Quality of Service by means of technologies such as the Integrated Services architecture and the Differentiated Services architecture. Commonly, these are used in Access Networks and Transport Networks respectively.

When Integrated Services architecture supports Quality of Service within an Access network, it is necessary to establish Resource Reservations over the packets path. Then, one special situation occurs when, in MIPv6 networks or HMIPv6 networks, packets are forwarded over a tunnel between two points in the path (between the Home Agent and the Mobile Node in MIPv6 networks, or between the Mobility Anchor Point and the Mobile Node in HMIPv6 networks). In this scenario, each Resource Reservation within Tunnel Interior routers is ignored because routers distinguish reservations by means of the Five-tuple (the Five-tuple consists of IP Source Address, IP Destination Address, the Source port, the Destination port and the Protocol ID), but it is not carried in the external IP header of encapsulated packets. On the contrary, the Five-tuple only is carried within the original IP header. One solution to this problem for the Standard IntServ is described in RFC 2746 [3]. However, this approach resolves the problem at the Transport Layer level. Thus, it uses UDP encapsulation and distinguishes reservations by means of the UDP source port to identify such reservations. On the other hand, one new approach, named IntServ6, has appeared recently to support Quality of Service in wireless and wired IPv6 access networks. This proposal distinguishes Reservations within Tunnels by means of the Flow Label field of the IPv6 header and resolves the "Reservations within Tunnels" problem at the Network Layer level. IntServ6 was described in [4] and several parameters were evaluated for this approach. Also, in [5], the Packet Delay was evaluated under mobile scenarios for IntServ6 routers.

The present paper's focus is the evaluation of a new aspect of IntServ6, namely the performance of this proposal under the "Reservations within Tunnels" scenario. Then, we compare IntServ6 with the standard IntServ solution. Moreover, we will show that the Served Traffic at the routers can be improved by means of this new approach. To perform our study, we will use models for Served Traffic upper bound at the GPS scheduling process within a router. These models are based on EBB (Exponentially Bounded Burstiness) theory [6].

II. MATHEMATICAL MODEL FOR SERVED TRAFFIC

As was described before in [4], one Integrated Services router is composed by several modules such as, Control Admission Scheme, Classification process and packet scheduling process. During the forwarding packet process, each arriving packet is processed first in the classification module. The classification procedure verifies that the packet belongs to an established reservation and it sends the packet to the next step, the scheduling process. Finally, the scheduling process guarantees the committed bandwidth for each session, one of the main parameters to measure the QoS. The packet scheduling algorithm used in the standard Integrated Services architecture is the Weighted Fair Queuing (WFQ) approach. It has two versions, one for fluid models, named the Generalized Processor Sharing (GPS) method, and the GPS packetized approximation, named Packet-by-Packet Generalized Processor Sharing (PGPS) [7].

* This work was supported by the ERDF and the Spanish Government through the TIC2006-04504 project.

A. *Served Traffic in a GPS Server*

Generalized Processor Sharing generally assumes a fluid model (i.e., packets are infinitely divisible) and is work-conserving. Consider a GPS server with rate *r* serving *N* sessions. Following Parekh and Gallager's definition [7], each session *i* is assigned a fixed real-valued positive parameter ϕ_i, where $\{\phi_i\}_{1 \leq i \leq N}$ is called a GPS parameter. A session is backlogged throughout an interval if there are always queued bits of traffic for that session at all times in the interval $[\tau,t]$ [7]. When session *i* is backlogged, it is guaranteed a backlog clearing rate (or equivalently, a guaranteed amount of service per unit time) of

$$g_i = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r \tag{1}$$

For simplicity, we will assume *r*=1 throughout the paper.

The model proposed by Zhang in [8] studies the sample path behavior of the sessions served by a single GPS server. In that study, the traffic for each arriving process A_i, is characterized by an E.B.B process with Long Term Upper Rate ρ_i such that $\Sigma^{N}_{i=1} \rho_i < 1$.

By abuse of notation, for $1 \leq i \leq N$, let A_i denote a sample path (or a realization) of a random arrival process A_i. Also, let $A_i(\tau,t)$ denote the amount of traffic from session *i* over a time interval $[\tau,t]$ in this sample path. In general, it is assumed that A_i is a right continuous function with left limit, therefore $A_i(\tau,t)=0$ for any t. Similarly, S_i is used to denote the corresponding random process S_i (Departure process for session *i*).

The GPS server behavior for session *i* can be described by means of the Fig. 3. As we mentioned above, ρ_i is the Long Term Average Rate of session *i* traffic. Also, ρ'_i is the Expected Service Rate the server will eventually commit to session *i*. The dashed curve is the Expected Service S_i. The difference between A_i and the Expected Service curve is the Expected Backlog $\delta_i(t)$. Additionally, in Fig. 3, we can observe that between τ and *s*, the Expected Amount of Service catches up to the number of the arrivals, and thus the Expected Backlog vanishes.

Under this environment, the approach in [8] obtained several expressions for upper bounds of parameters such as the backlog, the delay and the served traffic of a session *i*. However, for our proposals, we take only the served traffic upper bound expression. That is,

$$S_i(\tau,t) \leq A_i(\tau,t) + \delta_i(\tau) + \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} \sum_{j=1}^{i-1} \delta_j(\tau) \tag{2}$$

Where the equation part given by $\frac{\phi_i}{\sum_{j=1}^{N} \phi_j} \sum_{j=1}^{i-1} \delta_j(\tau)$, is the traffic portion to serve for the (*i-1*) remaining sessions at the backlog clearing rate.

B. *Classification Effect over the Served Traffic*

In this paper, we compare two technologies for QoS support. These are the Standard IntServ and a new approach named IntServ6 proposed in [4, 5].

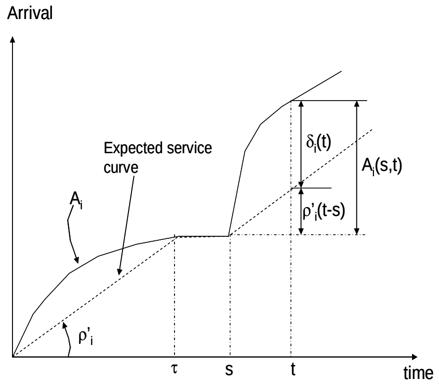

Figure 3. The GPS Server traffic behavior for session i.

The main difference between them is the packet classification process, whereas the scheduling discipline is the same for both approaches. Consequently, there are no direct effects over each served session bandwidth due to different scheduling schemes.

However, it is necessary to consider that changes in classification process have consequences over the Average Classification Time (ACT or X_c). As a result, changes in the ACT produce indirect effect over the amount of served traffic by the scheduling process when the scheduler is within a non-saturation state. We can assume that, if the ACT is reduced then the Mean Packet Arrival Rate (MPAR or λ) to the scheduling process will increase because $\lambda=1/X_c$. Moreover, such MPAR increases will be the same for all the packets arriving to the classification process. These assumptions can be stated due to both modules (classification and scheduling) being statistically independent. This can be explained because the service discipline in the classification module only processes the IPv6 header; then, the service time does not depends on the packet length. On the other hand, the scheduling service time depends on the packet length.

Under this environment, if we suppose that the MPAR increases by a factor of *p*, also we can conclude that the input traffic A_i for each session between times τ and *t* will increase in the same factor. This is,

$$A'_i(\tau,t) = p\, A_i(\tau,t) \tag{3}$$

Similarly, we can assume that the Queue Backlog for all the sessions will increase in the same proportion *p* as follows,

$$\delta'_i(\tau) = p\ \delta_i(\tau) \tag{4}$$

Then, we can make changes to expression (2) to obtain the expression (5) that describes the new served traffic upper bound as follows,

$$S'_i(\tau,t) \leq pA_i(\tau,t) + p\delta_i(\tau) + \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} p \sum_{j=1}^{i-1} \delta_j(\tau) \tag{5}$$

As an important result, from (5) we can conclude that if we increase the MPAR by a factor of *p*, then the served traffic upper bound for each session will increase in the same factor *p*. Then, *p* is a parameter measuring the improvement in the served traffic of a GPS server when the MPAR increase. We will name *p* as the "Served Traffic Gain".

The way to calculate the parameter p is to calculate the proportion between the original MPAR (λ) and the new MPAR (λ')as follows,

$$p=\lambda'/\lambda \tag{6}$$

$$p=(1/X_c')/(1/X_c) \tag{7}$$

Finally,

$$p=X_c/X_c' \tag{8}$$

Expression (8) will be used to calculate the improvement proportion in the Served Traffic. For our proposals, X_c' is the mean classification time of IntServ6 and X_c is the mean classification time of IntServ.

III. Clasification Service Time Model

To compare both QoS technologies (IntServ and IntServ6) over two different scenarios such as Simple Reservations and Reservations within Tunnels, we have obtained a stochastic model for the classifier mean service time for each situation. We refer to Simple Reservations as reservations over one typical path in the Internet. On the other hand, there are Reservations within Tunnels, which are sessions that reserve resources in routers over a tunnel path. Our model for the mean classification time is based on the states transition diagram (STD) of the process (see Fig. 4).

A. Simple Reservations Case

In Fig. 4, we observe the STD for both QoS technologies under a simple reservation environment. Some key 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, while in contrast this process in IntServ uses a matching scheme (see [9]). 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 (see [4]) is,

$$X_c= t_q + t_h + mt_{comph} + t_e + t_{comp}\, mC_h^2 \tag{9}$$

And the expression for service time on IntServ6 (see [4]) is,

$$X_c'= t_i + t_e + t_{comp}\, mC_h^2 \tag{10}$$

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 [10] 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 (or sessions) on the router.

B. Reservations within Tunnels Case

The expression for the mean classification time in this environment is obtained from the STD in Fig. 4. The STD for the IntServ6 case is the same that the IntServ6 simple reservations case because the classification process within the tunnel is based on the flow label field. On the other hand, the STD for IntServ in Fig. 4 has one additional state. The main difference between both technologies is that IntServ need to analyze the UDP port to distinguish the reservation within the tunnel. Then, the mean classification time for IntServ6 uses the same expression (10), whereas for IntServ it is necessary to add to expression (9) the required time for the UDP port analysis. Following the same rules of the model described in [4, 5], this additional time can be calculated as the number of Static RAM accesses necessary to read the UDP port combined with the number of SRAM readings required to find the internal tunnel reservation within the Tunnel Reservation Table, which holds all the reservations within the tunnel.

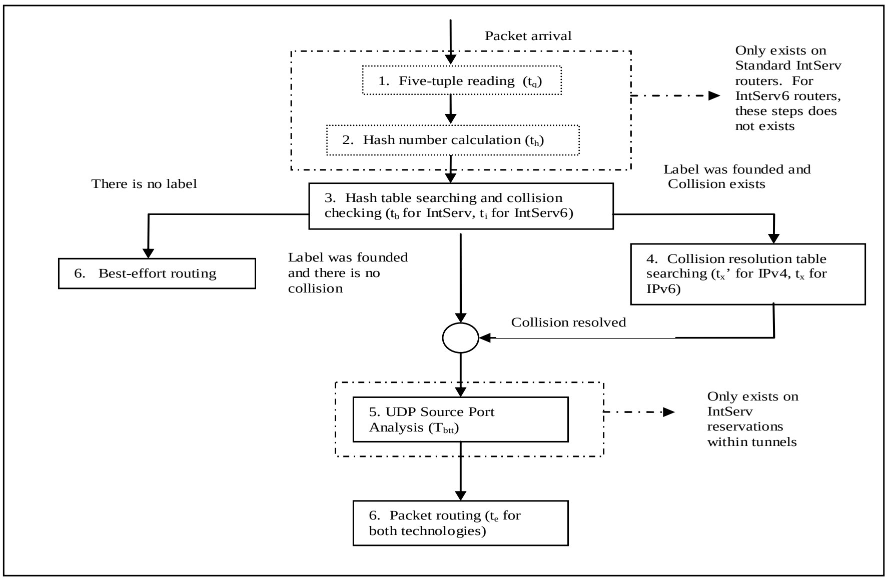

Figure 4. State transition diagram for the classification process on Integrated Services Routers

This search is matching type and, consequently, the stochastic model is similar to the Collision Resolution Table [4], i.e., it can be described by means of a geometric distribution. Then, the mean number of readings can be calculated as,

$$T_{btt}=m_{tunnel}T_{sram} \tag{12}$$

The expression for (12) then should be added to expression (9) to obtain the new expression for the IntServ tunneling mean classification time as follows,

$$X_c= X_{co} + m_{tunel}\, T_{sram} \tag{13}$$

Where X_{co} is given by expression (9).

IV. Evaluation Results

As was explained in section II, the *p* parameter is a measure of the improvement on the served traffic upper bound. By replacing expressions (9) and (10) in expression (8), we can obtain the served traffic improvement in the simple reservations case. In these expressions, we can change the number of simple reservations, *m*, and we observe the served traffic improvement behavior when *m* increases.

Processing times are calculated for the IXP 1200 Intel Network Processor, which was selected to follow the same conditions of other studies developed before in [4, 5]. The improvement in served traffic for IntServ6 with respect to IntServ in Simple Reservations can be observed in Fig. 5.

We can see that when the number of Simple Reservations increases, the improvement in the served traffic upper bound increases for IntServ6 respect to IntServ.

Now, if we replace expressions (10) and (13) in expression (8), we can describe the served traffic improvement in the reservations within tunnels case. Here, we can change two parameters, the number of simple reservations in the router (*m*) and the number of reservations within tunnels (m_{tunnel}). However, now we only study the effect of changes in the number of reservations within a tunnel (m_{tunnel}), so *m* will remains constant in m=200 simple reservations. Results can be observed in Fig. 6. Again, we observe that for IntServ6 the improvement in served traffic respect to IntServ increase when the number of reservations within tunnels increase.

This behavior occurs because for IntServ the mean searching time to distinguish reservations is proportional to the number of reservations within the tunnel, meanwhile, for IntServ6 the service time remains constant.

Figure 5. Served traffic improvement for IntServ6 with respect to IntServ in a simple reservations environment.

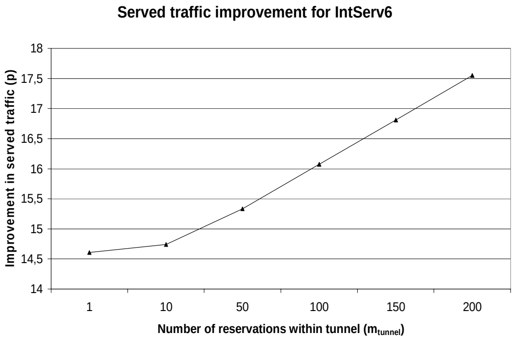

Figure 6. Served traffic improvement for IntServ6 with respect to IntServ in a “Reservations within tunnels” environment

IV. Conclusions

One important situation in All-IP Mobile Networks is the packet tunneling in MIPv6 and HMIPv6 networks. When Integrated Services architecture supports QoS in these networks, the current solution for Reservations within Tunnels over IntServ has some problems because it should distinguish reservations at the Transport Layer level. However, one new solution has emerged; it is named IntServ6. This approach manages reservations by means of the Flow Label field of the IPv6 packet header. Thus, it resolves the problem of reservations within tunnels management at the Network Layer level. In this paper, we verified by means of a stochastic model and simulation results (in MATLAB) that this approach reduces the packet processing time at the routers and hence this approach serves more traffic under simple reservations and reservations within tunnels.

References

[1] C. Perkins, "Mobility Support in IPv6," IETF, RFC 3775 2004.

[2] H. Soliman y C. Castelluccia, *Hierarchical Mobile IPv6 mobility Management (HMIPv6)*: IETF RFC 4140, 2005.

[3] A. Terzis, *RSVP Operation Over IP Tunnels*: IETF RFC 2746, 2000.

[4] J. Padilla y J. Paradells, "IntServ6: An approach to support QoS over IPv6 wired and wireless networks", *European Transactions on Telecommunications*, 2007.

[5] J. Padilla, J. Paradells, y A. Rodríguez, "Supporting QoS over IPv6 wireless networks with IntServ6," The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC’06), 2006.

[6] O. Yaron y M. Sidi, "Performance and Stability of Communication Networks via Robust Exponential Bounds", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 1993, 1993.

[7] A. K. Parekh y R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344-357, 1993.

[8] Z. Zhang, D. Towsley, y J. Kurose, "Statistical Analysis of Generalized Processor Sharing Sheduling Discipline," SIGCOMM'94, 1994.

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

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