

TIGHT ARRIVAL CURVE AT THE OUTPUT OF A WORK-CONSERVING BLIND MULTIPLEXING SERVER

Juan ECHAGÜE, Vicent CHOLVI

Departamento de Ingeniería y Ciencias de lo Computadores, Universitat Jaume I
Campus del Riu sec, 12071-Castellón-Spain
E-mail: Juan.Echague@icc.uji.es

Abstract. As a means of supporting quality of service guarantees, aggregate multiplexing has attracted a lot of attention in the networking community, since it requires less complexity than flow-based scheduling. However, contrary to what happens in the case of flow-based multiplexing, few results are available for aggregate-based multiplexing.

In this paper, we consider a server multiplexer fed by several flows and analyze the impact caused by traffic aggregation on the flows at the output of the server. No restriction is imposed on the server multiplexer other than the fact that it must operate in a *work-conserving* fashion.

We characterize of the best *arrival curves* that constrain the number of bits that leave the server, in any time interval, for each individual flow. These curves can be used to obtain the delays suffered by packets in complex scenarios where multiplexers are interconnected, as well as to determine the maximum size of the buffers in the different servers.

Previous results provide tight delay bounds for networks where servers are of the FIFO type. Here, we provide tight bounds for any work-conserving scheduling policy, so that our results can be applied to heterogeneous networks where the servers (routers) can use different work-conserving scheduling policies such as First-In First-Out (FIFO), Earliest Deadline First (EDF), Strict Priority (SP), Guaranteed Rate scheduling (GR), etc.

Key words: Performance evaluation, Aggregate scheduling, Blind multiplexing, Arrival curve.

1. Introduction. Although it is known that in flow-based architectures tight delay bounds can be computed for single flows, the computation overhead associated to per-flow management operations (i.e., classification and scheduling) is often unfeasible at high speeds when dealing with many flows simultaneously. For this reason, in large-scale packet-switched-networks, as the Internet, flows are managed as an aggregate rather than per-flow state operation. Examples of such architectures are *Differentiated Services* (Diff-Serv) adopted by the IETF (Blake1998) and *Multi-Protocol Label Switching* (MPLS) (Rosen2001).

Whereas most of the work done using aggregate multiplexing assume that aggregate packets are served using a FIFO discipline, in practice, many devices cannot be accurately described by a FIFO service model because packets arriving at the output queue from different input ports may experience different delays when traversing a node. This is due to the fact that many networking devices like routers are implemented using input-output buffered crossbars and/or multistage interconnections between input and output ports. Hence, packet reordering on the aggregate level is a frequent event and should not be neglected.

In this paper, we analyze the impact caused by traffic aggregation on the flows at the output of a work-conserving blind multiplexing server. Concretely, we a tight arrival curve which constrains the number of packets that leave the server, during any time interval. The paper is organized as follows. Section 2 describes our assumptions and notation. Section 3 shows our main result and it its shown its optimality (tightness). A review of related work is reported in Section 4. In Section 5, we show the suboptimality of taking an iterative approach to obtain the arrival curves. Finally, in Section 6, we present some conclusions and point out future work.

2. Aggregate scheduling: model and notation. We consider I flows which are served as an aggregate in a work-conserving blind multiplexing server with constant service rate R .

A fluid approach is taken, leaving packetization effects for further study; these effects are likely to have an impact on our results in the order of one maximum packet size (Chang2000, Le-Boudec2002).

We call $A_i(t)$ to the input function which computes the number of bits observed in flow i arriving to the server between the time interval 0 to t . Similarly, let $B_i(t)$ be the output function which computes the number of bits, for flow i , during the same time interval. Let $A(t) = \sum_{i=1}^I A_i(t)$ be the aggregate input function, and $B(t) = \sum_{i=1}^I B_i(t)$ the aggregate output function.

We assume that flow i arriving to the server is constrained by an arrival curve α_i (Cruz1995):

$$A_i(t) - A_i(s) \leq \alpha_i(t - s) \quad (1)$$

for all t, s such that $s \leq t$

Arrival curves are of great importance since are used to obtain the bounds on the delays suffered by packets, as well as properly dimension the buffer sizes (Cruz1995, Fidler2005, Le-Boudec2001, Lenzini2005, Ying2005).

Our problem, is to find the tightest arrival curves ($\alpha_i^*(x)$) for the output functions $B_i(t)$:

$$B_i(t) - B_i(s) \leq \alpha_i^*(t - s) \quad (2)$$

for all t, s such that $s \leq t$

In other words, we look for the tight arrival curve which bounds flow i at the output of the work-conserving blind multiplexing server. Note that this arrival curve will be used to bound the flow at the input of the next server on flow's path.

We focus on the case where arrival curves are *leaky-bucket* constrained. This is a typical constraint, which allows for bursts of a certain size and a defined sustainable rate (Cruz1995, Jiang2002, Cholvi2002, Fidler2005, Le-Boudec2001, Lenzini2005, Ying2005). More formally, the arrival curve that is enforced by a leaky-bucket with depth b_i and sustainable rate r_i is the function

$$\alpha_i(x) = b_i + r_i x \quad (3)$$

for all $x \geq 0$

An interesting feature of leaky-bucket curves is that they allow for a very simple description of traffic aggregation, since the aggregate arrival curve of a number of flows is the sum of the individual arrival curves.

PROPOSITION 1. The aggregate of a set of leaky-bucket constrained curves I according to Equation 3 is a single leaky-bucket curve given by

$$\alpha_i(x) = \sum_{j \in I} b_j + \sum_{j \in I} r_j x \quad (4)$$

for all $x \geq 0$

Therefore and without loss of generality, we can focus on flow $i = 1$ and consider the other flows as one aggregate flow. Thus, we can limit ourselves to the case $I = 2$ and find an arrival curve for the output of flow 1 (flow 2 represents the aggregate of all flows other than 1).

The term *scenario* is understood to mean any arbitrary collection of functions $(A_i(t))_{1 \leq i \leq I}$ that are wide-sense increasing and non-negative, and each function satisfies Equation 1. For convenience, whenever necessary, we use a

super-index to identify a scenario. For example, $B_i^\gamma(t)$ is the output function of flow i corresponding to scenario γ .

Let Γ be the set of all scenarios. Our problem is now to find the best possible arrival curve $\alpha_1^*(x)$ for the output flow $B_1(t)$. In other words, we should have, for any scenario $\gamma \in \Gamma$:

$$B_1^\gamma(t) - B_1^\gamma(s) \leq \alpha_1^*(t - s) \quad (5)$$

for all t, s such that $s \leq t$

We say that the arrival curve α_1^* is tight if it is the smallest possible one. Whenever necessary, we will use a super-index to identify the particular work-conserving policy used to perform the aggregation of flows (e.g., Γ^P is the set of all scenarios in a system using P as scheduling policy).

Call $B_{req} := \sup_{x \geq 0} [\alpha_1(x) + \alpha_2(x) - Rx]$ the worst case buffer required for a loss-free operation. We assume the finiteness condition

$$B_{req} < \infty. \quad (6)$$

Otherwise, it can easily be seen that our problem has no finite solution. A necessary and sufficient condition for a finite buffer requirement is that $r_1 + r_2 < R$, assuming that $b_1 + b_2$ is finite.

3. Optimal arrival curve at the output flow. In this section, we obtain the result that gives the solution to our problem. Before we proceed with it, we will introduce some new concepts. Consider some arbitrary but fixed time interval $[s, t]$.

- Denote as a the value of $x \geq 0$ that solves the following equation:

$$\alpha_2(x) - Rx = 0. \quad (7)$$

Since α_2 is a leaky-bucket function and $r_2 < R$ then a is defined and unique; namely, $a = \frac{b_2}{R-r_2}$. Denote $s_a = s - a$. Also, for a given scenario $\beta \in \Gamma$, denote as s^β the start of the busy period¹ that lasts, at least, until s .

- For any given scenario $\beta \in \Gamma$, denote as $q^\beta(s)$ the number of bits of flow 1 in the queue at time instant s .

¹A *busy period* is a period where the server buffer is non-empty.

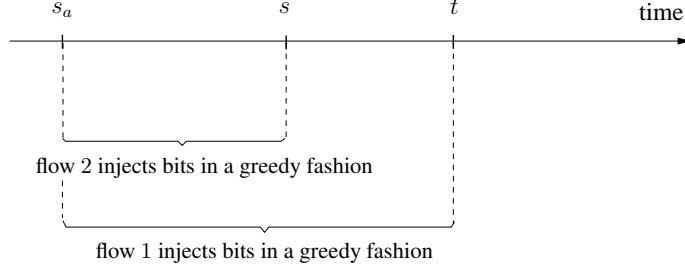


Fig. 1. Scenario $\phi(s, t)$ showing the amount of flow 1 and flow 2 bits injected in different time intervals.

- We say that scenario $\beta \in \Gamma$ *confines* scenario γ for time interval $[s, t]$, denoted as $\gamma \prec_{[s,t]} \beta$, if $B_1^\gamma(t) - B_1^\gamma(s) \leq B_1^\beta(t) - B_1^\beta(s)$. We also say that β is the *confiner* for time interval $[s, t]$ if $\forall \gamma \in \Gamma (\gamma \prec_{[s,t]} \beta)$.
- Given a scenario $\beta \in \Gamma$, we define β^g as the scenario obtained from β making flow 1 to inject in time interval $[s - \epsilon, t]$ (taking ϵ so that bits injected at time instant $s - \epsilon$ are transmitted at or after time instant s) the same number of bits injected by flow 1 in the same interval in β , but in a greedy fashion².
- Define *Second Flow First (SFF)* as the work-conserving policy that gives preference to bits from the second flow. Define $\phi(s, t)$ as the scenario in Γ^{SFF} such that (see Fig. 1):
 - In time interval $[s_a, t]$ flow 1 injects $\alpha_1(t - s_a)$ bits in a greedy fashion.
 - In time interval $[s_a, s]$ flow 2 injects $\alpha_2(s - s_a)$ bits in a greedy fashion.
 - In the rest of the intervals, no flow injects any bits.

Regarding the form of the output function for flow 1 corresponding to scenario $\phi(s, t)$, we have the following result:

Lemma 1. *For any given time interval $[s, t]$, we have that $B_1^{\phi(s,t)}(t) - B_1^{\phi(s,t)}(s) = \min\{R(t - s), \alpha_1(t - s_a)\}$.*

Proof. Taking into account how $\phi(s, t)$ has been defined, we have that, at time instant s , all bits injected by flow 2 will be transmitted, and the buffer will contain $\alpha_1(s - s_a)$ bits from flow 1.

²We say that in scenario β flow 1 injects bits in a greedy fashion in time interval $[s, t]$ if $\forall m : s \leq m \leq t (A_1^\beta(m) - A_1^\beta(s) = \alpha_1(m - s))$ (resp. for flow 2).

- Case $R(t-s) > \alpha_1(t-s_a)$: This means that not all injected flow 1 bits have been transmitted at time instant t and, since at time s all injected flow 2 bits have been transmitted, then $R(t-s)$ will be transmitted at time t .
- Case $R(t-s) \leq \alpha_1(t-s_a)$: Since $R(t-s) \leq \alpha_1(t-s_a)$ then, at time t the buffer will be empty. This means that all injected flow 1 bits have been transmitted. Namely, $\alpha_1(t-s_a)$.

Furthermore, we have that $\phi(s, t)$ is the scenario that provides the highest number of flow 1 bits in the queue at time instant s .

Lemma 2. *For any given time interval $[s, t]$, we have that $\forall \beta \in \Gamma (q^\beta(s) \leq q^{\phi(s,t)}(s))$.*

Proof. If $s^\beta \geq s_a$ the proof is immediate. Therefore, assume that $s^\beta < s_a$. Proof by contradiction. Assume that there is a scenario $\beta \in \Gamma$ such that $(q^{\phi(s,t)}(s) \leq q^\beta(s))$. Therefore, we have that $\alpha_1(s-s^\beta) + \alpha_2(s-s^\beta) - R(s-s^\beta) > \alpha_1(s-s_a)$. Taking into account that both α_1 and α_2 are leaky-bucket constrained, we have that $b_1 + r_1(s-s_a) + r_1(s_a-s^\beta) + b_2 + r_2(s-s_a) + r_2(s_a-s^\beta) - R(s-s_a) - R(s_a-s^\beta) > b_1 + r_1(s-s_a)$, which implies that $r_1(s_a-s^\beta) + b_2 + r_2(s-s_a) + r_2(s_a-s^\beta) - R(s-s_a) - R(s_a-s^\beta) > 0$. Since $b_2 + r_2(s-s_a) - R(s-s_a) = 0$ then $r_1(s_a-s^\beta) + r_2(s_a-s^\beta) - R(s_a-s^\beta) > 0$, and consequently $(r_1 + r_2)(s_a-s^\beta) > R(s_a-s^\beta)$. Clearly, this implies that $r_1 + r_2 > R$. But, by definition, $r_1 + r_2 \leq R$, thus contradicting our initial hypothesis.

The previous lemma provides us with an upper bound on the maximum number of flow 1 bits queued at any time instant.

COROLLARY 1. For all $\beta \in \Gamma (\max_{\forall s} q^\beta(s) = \alpha_1(a))$.

Proof. Consider any time instant s . We know, from Lemma 2, that $\forall \beta \in \Gamma (q^\beta(s) \leq q^{\phi(s,t)}(s))$. But since $q^{\phi(s,t)}(s) = \alpha_1(a)$ (see how $q^{\phi(s,t)}(s)$ has been defined) then the proof follows.

The following lemma relates the "confinement" relationship of any scenario $\beta \in \Gamma$ with its "greedy" version (i.e., with β^g).

Lemma 3. *For any given time interval $[s, t]$, we have that $\forall \beta \in \Gamma (\beta \prec_{[s,t]} \beta^g)$.*

Proof. Immediate, taking into account that, in time interval $[s-\epsilon, t]$, the number of injected bits in both scenarios will be the same and in β^g they have been injected in a greedy fashion.

Now, we can state the main theorem.

Theorem 1. *Consider a system serving two flows in an arbitrary aggregate manner, with the assumptions in Section 2. Define*

$$\alpha_1^*(x) = \min\left\{Rx, \alpha_1\left(x + \frac{b_2}{R - r_2}\right)\right\}.$$

Then α_1^* is the tight arrival curve for the output flow $B_1(t)$ that can be found under these assumptions.

Proof. First, we prove that for any given time interval $[s, t]$, we have that $\forall \beta \in \Gamma$ then $\beta \prec_{[s, t]} \phi(s, t)$.

- **Case $s^\beta \geq s_a$:** Consider scenario $\beta \in \Gamma$. On the one hand, given a time interval $[s, t]$, the buffer cannot transmit at a rate that is higher than R (i.e., $R(t - s)$). Furthermore, the buffer cannot transmit, in time interval $[s, t]$, more than what is injected in time interval $[s^\beta, t]$, which is, at most, $\alpha_1(t - s^\beta)$. Therefore, we have that $B_1^\beta(t) - B_1^\beta(s) \leq \min\{R(t - s), \alpha_1(t - s^\beta)\}$. Now, taking into account Lemma 1 and the fact that $s^\beta \geq s_a$, we have that $B_1^\beta(t) - B_1^\beta(s) \leq \min\{R(t - s), \alpha_1(t - s^\beta)\} \leq \min\{R(t - s), \alpha_1(t - s_a)\} = B_1^{\phi(s, t)}(t) - B_1^{\phi(s, t)}(s)$. Consequently, $\beta \prec_{[s, t]} \phi(s, t)$.
- **Case $s^\beta < s_a$:** On the one hand, the number of bits that can be transmitted in time interval $[s, t]$ in scenario $\phi(s, t)$ is $\min\{q^{\phi(s, t)}(s) + r_1(t - s), R(t - s)\}$. Similarly, in scenario β^g we have that it is bounded by $\min\{q^{\beta^g}(s) + r_1(t - s), R(t - s)\}$. Since, from Lemma 2, $q^{\beta^g}(s) \leq q^{\phi(s, t)}(s)$ then $\min\{q^{\beta^g}(s) + r_1(t - s), R(t - s)\} \leq \min\{q^{\phi(s, t)}(s) + r_1(t - s), R(t - s)\}$. Therefore $\beta^g \prec_{[s, t]} \phi(s, t)$. Furthermore, from Lemma 3, we have that $\forall \beta \in \Gamma$ ($\beta \prec_{[s, t]} \beta^g$). Then $\beta \prec_{[s, t]} \phi(s, t)$.

Therefore, we have that the scenario $\phi(s, t)$ is the confiner for time interval $[s, t]$. Taking into account Lemma 1 and considering how s_a has been defined, the proof follows.

Since α_1 is a leaky-bucket curve, then we have that α_1^* is bounded by the server rate R and by a leaky-bucket function with parameters (b_1^*, r_1) , with

$$b_1^* = b_1 + r_1 \frac{b_2}{R - r_2}.$$

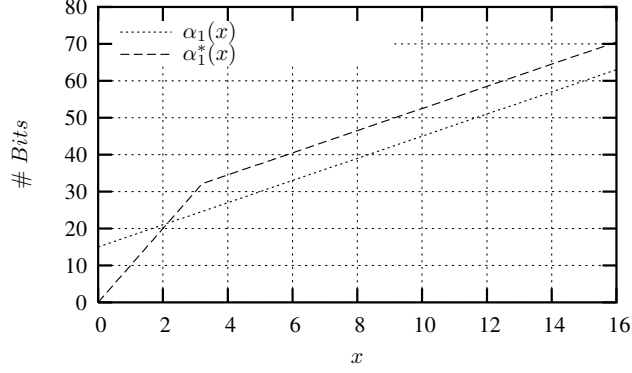


Fig. 2. Worst case arrival curves for the output of flow 1. The server rate is $R = 10$, $\alpha_1(x) = 15 + 3x$ and $\alpha_2(x) = 10 + 6x$.

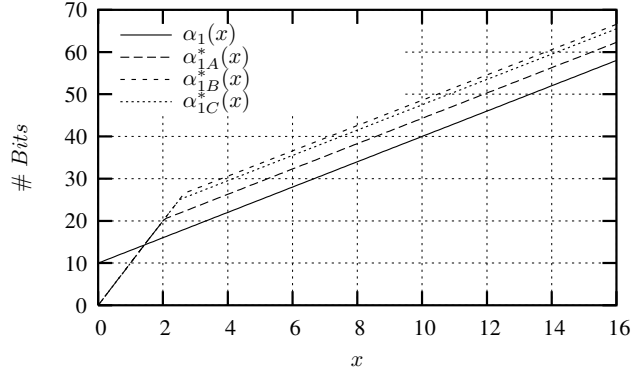


Fig. 3. Worst case arrival curve for the output of flow 1 when varying the parameters of the aggregate flow 2. The server rate is $R = 10$, $\alpha_1(x) = 10 + 3x$, $\alpha_{2A}(x) = 10 + 3x$, $\alpha_{2B}(x) = 20 + 3x$ and $\alpha_{2C}(x) = 10 + 6x$. $\alpha_{1i}^*(x)$ represents the arrival curves for the output of flow 1 when flow 2 is constrained by $\alpha_{2i}(x)$.

Fig 2 provides a numerical example of the shape of this curve. In Figure 3, we also illustrate how the parameters of flow 2 affect $\alpha_1^*(x)$. As expected, the increment of the sustainable ratio (r_2), as well as the burst (b_2) of the curve for flow 2, produces an increase in $\alpha_1^*(x)$.

4. Related work. Regarding previous work that has dealt with the characterization of arrival curves that constrain the number of packets that leave an arbitrary blind multiplexing server, to our knowledge, the only source that we are aware is a result in (Le-Boudec2001). In this result, when two leaky-bucket constrained flows ($\alpha_i(x) = b_i + r_i x$) arrive to a server with strict service curve $\beta_{R,T}$, when $r_1 + r_2 \leq R$ the output of flow 1 is constrained by a the following leaky-bucket arrival curve:

$$\alpha_1^{**}(x) = (b_1 + b_1 T + r_1 \frac{b_2 + r_2 T}{R - r_2}) + r_1 x$$

In a constant rate server, as assumed in this paper, $T=0$ and Equation 4 matches our result. Contrary to our result in (Le-Boudec2001) it is not demonstrated that the bound is optimal either for constant servers nor for a server providing a $\beta_{R,T}$ service curve.

A similar result has been reported in (Cruz1998) for the case of FIFO multiplexing (and extended for concave piecewise linear arrival curves in (Cholvi2002)). In that case, the arrival curve for the output of flow 1 (which is also optimal) is bounded by a leaky-bucket function with parameters (d_1, r_1) , with:

$$d_1 = b_1 + r_1 \frac{b_2}{R}$$

Now, we have that d_1 is lower than b_1^* . However, it must be taken into account that FIFO is a special case of work-conserving aggregate multiplexing policy (i.e., the FIFO bound is not valid, in general, for work-conserving multiplexing policies).

5. Feasible scenarios. In this section, we show that, contrary to what happens at the input, at the output not all valid scenarios in accordance with the worst case arrival curves can occur.

PROPOSITION 2. Consider a work-conserving server serving two flows (with the assumptions in Section 2). Then, at the output, not all valid scenarios in accordance with the worst case arrival curve can occur.

Proof. By counter-example. Let us focus on a system where $\alpha_1(x) = \alpha_2(x) = \min\{Rx, b + rx\}$. Clearly, we have that $\alpha_1^*(x) = \alpha_2^*(x)$. Denote as x' the point where α_1^* (and consequently α_2^*) changes the value of its linearity. Therefore $Rx' \leq b^* + rx'$.

Take a scenario, denoted β^* , in which both flow 1 and flow 2 are greedy in time interval $[s, s + x']$ at the output. By definition of the arrival curve (see

Equation 1), β^* is a valid scenario. Assume, by way of contradiction, that this scenario can occur. Since $Rx' \leq b^* + rx'$ then both flow 1 and flow 2 will transmit (at the output of the server) in time interval $[s, s + x']$ at a rate of R . This implies that, in time interval $[s, s + x']$, the server will transmit at a rate of $2R$, which is not possible.

As a consequence of this, we have that taking an iterative approach to characterize the arrival curves at the output when flows pass throughout several servers does not guarantee that they will be optimal (in the sense that, although valid, they do not necessarily have to be the tight arrival curves that can be found).

6. Conclusions. In this paper, we have analyzed the impact caused on the input flows by a work-conserving blind multiplexing server. We provide the tight arrival curve at the output of the server. Furthermore, it has been shown that the scenarios at the output of such server are, in general, more restrictive than those at the input, which explains the well-known inefficiency involved in finding performance bounds by iteratively applying output burstiness bounds in arbitrary work-conserving servers (Le-Boudec2001).

An important issue is to extend our result to the case where the arrival curve constraints for the input flows are piecewise linear functions, which correspond to combination of leaky-buckets. This will allow us to work directly with the variable bit rate case (or T-SPEC) used by the IETF.

REFERENCES

- R.L. Cruz. *Quality of Service Guarantees in Virtual Circuit Switched Networks*. IEEE JSAC, 1995.
- S. Blake and D. Black and M. Carlson and E. Davies and Z. Wang and W. Weiss. *An Architecture for Differentiated Services*. IETF RFC 2475, 1998.
- E. Rosen and A. Viswanathan and R. Callon. *Multiprotocol label switching architecture*. IETF RFC 3031, 2001.
- J-Y Le Boudec, and P. Thiran. *Network Calculus*. Springer Verlag Lecture Notes in Computer Science volume 2050 (available online at <http://icawww.epfl.ch>), 2001.
- C.S. Chang. *On deterministic traffic regulation and service guarantee: A systematic approach by filtering*. IEEE Transactions on Information Theory, 44, August 1998.
- M. Fidler and V. Sander and W. Klimala. *Traffic shaping in aggregate-based networks: implementation and analysis*. Computer Communications, 28, 2005.
- L. Lenzini and E. Mingozzi and G. Stea. *Delay bounds for FIFO aggregates: a case study*. Computer Communications, 28, 2005.
- Y. Jiang. *Delay bounds for a network of guaranteed rate servers with FIFO aggregation*. Computer Networks, 40, 2002.
- C.S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.
- J-Y Le Boudec. *Some Properties Of Variable Length Packet Shapers*. ACM/IEEE Transactions on Networking, 3, 2002.
- V. Cholvi and J. Echagüe and J-Y. Le Boudec. *Worst Case Burstiness Increase due to FIFO Multiplexing*. Performance Evaluation, 49, 2002.

- Y. Ying and R. Mazumdar and C. Rosenberg and F. Guillemin. *The Burstiness Behavior of Regulated Flows in Networks*. IFIP Networking, 2005.
- R. L. Cruz. *SCED+ : Efficient Management of Quality of Service Guarantees*. IEEE Infocom, San Francisco, 1998.

Received February 2008

J. Echagüe received the Ph.D degree in Computer Sciences at the Jaume I University (Spain). Currently is teaching as an assistant professor. His current interests are related to provide with quality of service to broadband networks, and routing protocols in MANETS.

V. Cholvi graduated from the University of Valencia (Spain) and received his doctorate in 1994 from the Polytechnic University of Valencia (Spain). In 1995, he joined the Jaume I University in Castelló where he is currently an associate professor. He is in charge of the Distributed Systems Group and his interests are in distributed and communication systems.