One-Hop Delay Estimation in 802.11 Ad Hoc Networks Using the OLSR Protocol

Amina Meraihi Naimi, Philippe Jacquet

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

One-Hop Delay Estimation in 802.11 Ad Hoc Networks Using the OLSR Protocol

Amina Meraihi Naimi, Philippe Jacquet

N° 5327

Octobre 2004

THÈME 1

Rapport de recherche

ISSN 0249-6399

One-Hop Delay Estimation in 802.11 Ad Hoc Networks Using the OLSR Protocol

Amina Meraihi Naimi, Philippe Jacquet

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n° 5327 — Octobre 2004 — 12 pages

Abstract: In this report, we present an estimation of the one-hop delay in 802.11 wireless network using the OLSR routing protocol. Our approach consists in modeling the 802.11 Mac sublayer while taking advantage of statistics obtained through the use of OLSR. We model each node as a discrete time M/G/1 queue. The Mac service time distribution is derived by modeling the exponential backoff function and deducing the collision probability and the channel occupancy from OLSR statistics. Our model is verified through simulations using the ns-2 tool.

Key-words: One-hop delay, OLSR, 802.11, M/G/1, collision probability.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Estimation du délai à un saut dans un réseau ad hoc 802.11 en utilisant OLSR

Résumé : Dans ce rapport de recherche, nous proposons une nouvelle méthode d'estimation du délai à un saut dans un réseaux ad hoc 802.11 en utilisant le protocole de routage OLSR. Notre approche consiste à combiner la modélisation de la couche Mac 802.11 et les statistiques obtenues grâce au protocole OLSR. Un noeud mobile est modélisé par un système M/G/1 à temps discret, ainsi le délai moyen du noeud est fonction du taux d'arrivées des paquets dans la file d'attente et la distribution du temps de service, dans ce cas le temps de traitement à la couche Mac . La distribution du temps de service est dérivée en modélisant la procédure exponentielle du backoff et en déduisant la probabilité de collision et l'occupation du canal grâce aux statistiques obtenues par le biais de l'utilisation des paquets Hello du protocole OLSR. Notre modèle a été testé et validé a travers des simulations ns-2.

Mots-clés : Délai à un saut, OLSR, 802.11, M/G/1, Probabilité de collision.

1 Introduction

With the emergence of real-time applications in mobile ad hoc networks (Manet), delay guarantees are increasingly required. In order to optimize delay in such networks, it is necessary to be able to evaluate it in an accurate way. We propose in this work to estimate the one-hop delay in ad hoc networks. As node delay depends greatly on the routing and the Mac layer protocols used, we consider the particular case of the OLSR ad hoc routing protocol [1]and the IEEE 802.11 Mac protocol. OLSR is a proactive routing protocol based on the use of periodic control packets. Mainly, Hello and Tc packets which are used to maintain the neighbor and the topology table respectively. The 802.11 protocol is considered to be the most popular Mac protocol in Manet. The 802.11 protocol can operate in two modes as defined in the standard[2]. The DCF mode, based on CSMA/CA, was designed for asynchronous traffic while, the PCF mode based on time division was designed for synchronous traffic. However, since PCF is no longer implemented, DCF is the only mode used for both types of traffic. As it is a random access protocol, DCF does not give any guarantee for delay sensitive applications. Many studies have been interested in modeling the DCF mode. Most of them focused on throughput and capacity[5][6]. Recently, some authors have studied the delay in particular[4][7]. We have noted that modeling the DCF access mode is not an easy task, as a lot of constrains and assumptions have to be considered. Our idea consists in partially modeling the access delay in an analytical way. We take advantage of the statistics obtained through the use of the OLSR routing protocol to complete the modeling. In [3], we developed an extension of the OLSR protocol to support delay. We presented a simple way to estimate the average one-hop delay and focused on the impact of the delay-oriented routing on improving the end-to-end delay. In this work, we propose a more accurate model and therefore focus on one-hop delay estimation.
The rest of the document is organized as follows. In section 2, we give an overview of the IEEE 802.11 DCF mode and present existing work related to the analytic modeling of the DCF mode. In section three, our model is presented. In section four, simulation results are shown and analyzed. Finally, a conclusion is given.

2 OVERVIEW OF THE 802.11 DISTRIBUTED COORDINATION FUNCTION : DCF

DCF is the fundamental access method used in 802.11. It is based on the CSMA/CA mechanism and characterized essentially by the following points:
The station transmits only if the medium is idle. The medium is considered to be idle if it is sensed idle for a duration called "DIFS" (Distributed Interframe Space). If the medium becomes idle after a busy period, collisions may occur because multiple stations could have been waiting for the current transmission and attempt to access the channel again at the same time. Therefore, a random period is introduced. It is called "backoff time", chosen in an interval called "contention window". In spite of that, a collision can occur, for example, when the chosen random backoff is the same for at least tow stations. In order to reduce

the probability of further collisions, the contention window is doubled after each collision to increase the random backoff and thus the waiting time. This function is called the "exponential backoff function". To make sure that the transmitted frame has reached its destination, an Acknowledgement frame is generated from the destination to the source.
The above carrier sense is called "physical carrier sense" because it is performed at the air interface. A virtual carrier sense is also possible in the DCF mode to resolve the problem of the hidden terminal. It is performed at the MAC sublayer. Here, the channel is reserved before each transmission. So instead of transmitting the data frame after sensing that the channel is idle, the station sends an RTS (Request to Send) frame to the destination. The receiver replies by a CTS (clear to Send) frame after which data transfer can start. Additional delay and bandwidth overhead are imposed by the use of RTS/CTS frames. Therefore, both modes "basic DCF" or "DCF with RTS/CTS" can be used.

A. 802.11 Delay Modeling : Related work

A great deal of work has recently been carried out in modeling the 802.11 protocol. Delay has been less studied than throughput but basically the modeling steps are similar. The modeling begins by determining the channel probabilities. We present below the most well known. All the models operate on a one-hop network. They suppose that the number of nodes is fixed and known (without mobility). Also, the collision probability is considered to be constant, and the same in all the nodes. In a real environment, collision probability is not constant as it is a function of the distance and the packet size.
Bianchi in [5] was interested in throughput. He considered a saturated network with a fixed number of stations in contention. Two parameters were mainly studied. p : collision probability, τ : the probability that a station transmits at a random slot. The key approximation in his work is the fact of considering p as constant i.e. independent of the number of retransmissions. The backoff timer is modeled using a "bidimensional Markov chain" : {backoff counter, backoff stage }. The stationary distribution is found and τ is deduced. Another important consideration in his work is that p corresponds to the probability that at a given slot, at least one of the remaining stations transmits. Hence, p is function of τ. By solving a non linear equation system, p and τ are found. A lot of studies have been based on these results.
In [7], authors were interested in delay. The two parameters mentioned above were used in the same way. The Markov chain was a little changed compared with [5]. The backoff counter was considered to be frozen either because of a successful transmission or because of a collision leading to differentiate two transition probabilities. The distribution of the service time is then determined. In [4] the delay is studied in both cases : saturated and non saturated networks. To do that, a new probability was considered. π_0: the probability that the system is non empty. Each node is modeled as a discrete time G/G/1 queue and the service time distribution is derived.

In order to estimate the one-hop delay in 802.11 ad hoc networks and avoid a constraining model which can not be applicable , we adopt a new approach. We model the 802.11 access time while we take advantage of the statistics obtained through the routing protocol, OLSR in our case. In the next section, we describe our delay modeling.

3 ONE-HOP DELAY ESTIMATION

A. M/G/1 Model

To obtain the average node delay, we model a mobile node using the 802.11 protocol as a discrete time M/G/1 queue. The average node delay represents the "response time" given by Little's law as :

$$W = \frac{\lambda M_2}{2(1-\lambda b)} + b \tag{1}$$

W =average response time, that is to say the one-hop delay,
b =average service time ; the time spent in the Mac layer. It corresponds to the time needed to successfully transmit the packet, or to discard it,
M_2 =second moment of the service time distribution,
λ =packet arrival rate.
We need to find the distribution of the service time to obtain b and M_2 by derivation.

B. Service Time Distribution

The Mac service time depends on how packets are transmitted i.e. with or without RTS/CTS (two or four handshaking). We study the first case and show how to deduce the delay if no RTS/CTS mechanism is used.

1. With RTS/CTS mechanism

As shown in the state diagram presented in Fig.1, data transmission time is equal to the time needed to successfully transmit an RTS packet (until the CTS packet is received), to which we add the time of transmission of the data packet. In a one-hop network, the hidden station and interference problem do not occur, so data packet collision probability can be neglected.

By using the z variable, data transmission time can be written as :

$$E[Z^{T_{data}}] = E[Z^{T_{rts}}]Z^{T_r} \tag{2}$$

T_r: distribution of transmission time of data packet, we consider it as constant.
T_{rts}: distribution of the successful transmission time of an RTS packet.
The transmission time of an RTS packet is the sum of the first transmission of the RTS packet and all the possible retransmissions in the case of collisions. The maximum backoff stage is equal to 5 according to the standard [2]. Above this limit, 2 retransmissions with

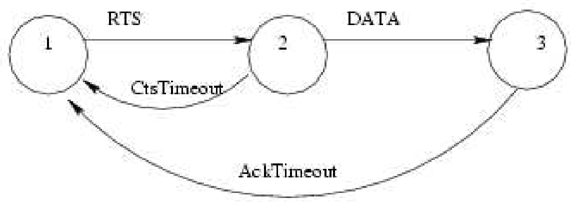

Figure 1: Data transmission

the same backoff window are allowed before discarding the packet. We consider as in [5], that all collisions occur with the same probability. We express T_{rts} distribution by :

$$E[Z^{T_{rts}}] = E[Z^{T_0}](1 - p_c + E[Z^{T_1}]p_c(1 - p_c + E[Z^{T_2}]p_c(1 - p_c$$

$$+\mathrm{E}[\mathrm{Z}^{T3}]p_c(1 - p_c + E[Z^{T4}]p_c(1 - p_c + E[Z^{T5}]p_c(1 - p_c$$

$$+\mathrm{E}[\mathrm{Z}^{T5}]p_c(1 - p_c + [Z^{T5}]p_c))))))) \qquad (3)$$

P_c :collision probability of RTS packets,
T_0 :distribution of the first transmission time for an RTS packet,
T_k :distribution of the K_{th} transmission time of an RTS packet.

Each transmission time of an RTS packet is equal to the channel access time to which we add "CTStimeout" which is constant. "CTStimeout" is equal to the time needed to transfer the RTS frame and to receive the CTS frame. We note the distribution of the K_{th} access time by Tac_k.

$$E[Z^{T_k}] = E[Z^{Tac_k}]Z^{CTStimout} \qquad (4)$$

Basically, when a data packet arrives at the MAC layer, 3 situations may occur :

- The channel is idle, the RTS packet is immediately transmitted. The access time is equal to "DIFS", approximately 2 slots.
- The channel is busy, a new random backoff is chosen and the RTS access time is the backoff time.
- The backoff procedure was started before the arrival of the packet. The RTS access time is the remaining backoff.

In our study, we consider that the first RTS access time corresponds to DIFS if the channel is found idle or to the backoff time if the channel is busy. The fact of neglecting the last situation can lead to a small over estimation of the access time. Recall that the access time of a retransmitted RTS packet always corresponds to the backoff process time (the second situation). As we know, the backoff is chosen uniformly in a contention window [Cwmin, Cwmax]. Each decrementation of the backoff requires one slot, if the channel is idle, or several slots if the channel is detected to be busy. The counter is frozen for a duration of a successful transmission or for a duration of a collision. In our study, we consider that the counter is frozen when the channel is busy without differentiating these two cases. This leads to considering only one random variable, called C_0, to represent the "channel occupancy". We express the access time by using a new operator $\otimes$ such as:

$$Tac_k = Uniform(1, 2^k Cwmin) \otimes C_0 \tag{5}$$

Which is equivalent to :

$$E[Z^{Tac_k}] = \frac{1}{2^k cwmin} E[Z^{C0}] + \frac{1}{2^k cwmin} E[Z^{C0}]^2 + \cdots + \frac{1}{2^k cwmin} E[Z^{C0}]^{cwmin} \tag{6}$$

C_0 :distribution of the number of slots spent for each decrementation of the backoff counter.

Let's note :
μ_{rts} :average RTS transmission time,
μ_{C_0} :average of C_0 distribution,
v_{rts} :second derivative of the distribution of RTS transmission time,
v_{C_0} :second derivative of C_0 distribution.

To obtain the first RTS access time, Tac_0 should be readjusted by taking into account the probability that a slot is idle, let's call it p_{idle}. We know that $p_{idle} = \frac{1}{\mu_{C_0}}$, then we have :

$$Tac_0 = (1 - p_{idle})Tac_0 + p_{idle}DIFS \tag{7}$$

By deriving (3), the average RTS transmission time is given by :

$$\mu_{rts} = \sum_{i=0}^{7} p_c^i (1 - p_c) \sum_{j=0}^{i} E[T_j] \tag{8}$$

From (4) and (6), we have the average K_{th} transmission

$$E[T_k] = \frac{\mu_{C_0}}{2}(2^k cwmin + 1) + CTStimout \tag{9}$$

Therefore, the average service time is given by :

$$\mu_{T_{data}} = b = \mu_{rts} + T_r \tag{10}$$

The second derivative of the RTS transmission time is obtained from (3) :

$$v_{rts} = \sum_{i=0}^{7} p_c^i (1-p_c) \prod_{j=0}^{i} (E[T_j^2] - E[T_j]) \tag{11}$$

The second moment of the service time M_2 is then given by :

$$M_2 = v_{rts} + \mu_{rts}(2T_r + 1) + T_r^2 \tag{12}$$

Based on this model, and having C_0 and p_c (see section C. below), the response time, that is to say the node delay, can be computed by replacing b and M_2 in Little's law.

2. Without RTS/CTS mechanism

The only thing to change is the equation (4). "CTStimeout" must be replaced by "Acktimout". "Acktimout" is equal to the time needed to transmit the data packet, that is to say T_r, to which we add the time to transmit the acknowledgement frame. The distribution of transmission time of the data packet is written in the same way as the distribution of transmission of an RTS packet as in (3).

C. Channel Occupancy and Collision Probability

C_0 and p_c are derived from the statistics obtained through the use of the OLSR protocol. We get C_0 by samples. Each time a Hello packet is sent, the backoff decrementation process is observed. The time needed for each decrementation is recorded. After the transmission of the Hello packet, the distribution C_0 is obtained as a vector, as shown in the example below :

1	1	1	1	277	1	1	1	1	1	1

Here, the chosen random backoff was 11. During the decrementation of the backoff, a transmission was detected (either successful or collision), the counter was frozen for 277 slots and then resumed .

As we notice, Hello packets serve to obtain knowledge of the channel occupancy, particularly if no data packet has been transmitted. The distribution C_0 is obtained based on one Hello packet. In order to get a better representation of the channel occupancy, we observe several Hello packets and concatenate the corresponding C_0 vectors. We assume that one could observe a Hello packet in the Mac interface. In future work, we intend to study the real possibilities.

Also, we use the Hello loss rate which is easily obtained from OLSR to approximate RTS collision probability. That is based on the fact that Hello packets as broadcasted, are lost in case of collisions. This approximation gives better results under the two following assumptions. First, as RTS collision corresponds to the Hello Mac loss, Hello loss in the queue should be negligible. Second, Hello packet size should not have any effect on collision probability. Since a Hello packet contains the list of all the one-hop neighbors, that implies that the number of nodes should be reasonable. Otherwise, Hello loss should be readjusted by taking into consideration the packet size, but this is out of scope of this work.

4 SIMULATIONS AND ANALYSIS

We use the ns-2[8] simulator to validate our delay modeling. We consider a one-hop ad hoc network with different number of nodes and load on the network. All the nodes are one-hop neighbors. We study various scenarios to compare the one-hop delay estimated (obtained by modeling) and the delay measured (obtained by finding the difference between the launching time and the arrival time). We also compare the RTS collision probability measured with the Hello loss rate. Characteristics of the simulation environment are summarized in Table 1.

In the first scenario, 5 neighbors are considered, each of which sends a CBR traffic. The launching time as well as the rate of the 5 sources were a little differentiated to avoid traffic synchronization which can lead to additional collisions. We use the word "throughput" to indicate the sum of all traffic rates. We measure the average delay under different throughputs and compute the global average delay. We compared it with the average estimated delay. First, we confirm the unfairness of the 802.11, i.e. delays are not the same in all the

Table 1: Simulation Settings

Routing protocol	OLSR
Simulation time	400s
Simulation Area	$1000\times1000m^2$
Bandwidth	2Mbs
Packet size	1000 bytes
Traffic	CBR

nodes, that is why we consider the average value of the delays of all the nodes. Second, by increasing the throughput, we note that the delay remains more or less steady and small (less than 20 ms) until a certain limit that we call a saturation limit. After which the delay increases highly. To show this variation, we consider throughput values just before and just after the saturation limit, as shown in Fig. 2.

In light and very light network, before the saturation limit which is equal to 1.564 mbps, we notice that the data estimated delay is a little greater than the delay measured. Under very low loss rates ($< 5\%$), that means that Hello packets delays were greater than RTS packets delays. As traffic sources were non-synchronized (traffic rates and launching times were different), data packets arriving at the Mac layer have few chance to contend with other data packets. On the other hand, as OLSR uses jitter before transmitting a Hello packet, the latter has few chance to contend with another Hello packet. Therefore, the only possibility of contention is between Hello and data packets. As data packet is bigger, Hello

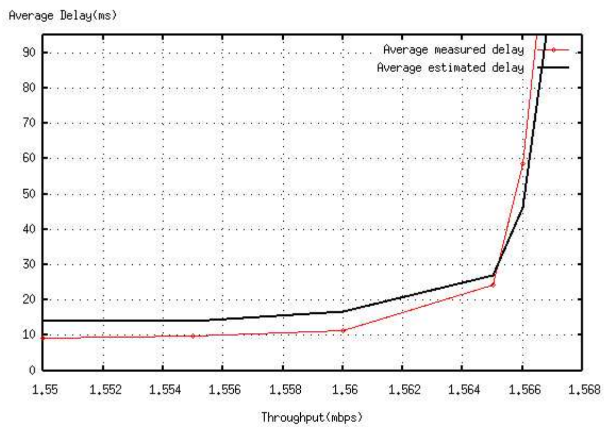

Figure 2: Estimated versus measured delay in 5 nodes network

packet waits more than data packet to access the channel. As we use Hello delays to estimate data packets delays, the delay is over estimated.
When the throughput increases highly (after the saturation limit), we first notice that the network is no more one-hop network. In another word, multi-hop routes due to links breaks can be selected within the network. As shown in Fig. 2, the average delay estimated and measured are close too. However, the latter is a little bit greater than the former. This can be explained by too many factors and different situations:

- Suppose that one RTS packet waits for long time to access the channel. This increases the waiting time and thus the one-hop delay for all packets in the queue, but not necessarily their Mac service time. Unfortunately, as we consider only the Hello Mac service time to obtain the RTS one and to deduce (little's law) the queue waiting time, this peak can not have an effect on the delay estimation.
- Packets are not source-routed. Therefore the route envisaged for packets is not necessarily the one followed and so for the delays.
- Route loss implies that there is no entry in the routing table, so corresponds to a "non-information" in term of estimation of delay which can effect the mean value.

We run the same simulations by considering 10 and 20 neighbors, with 10 and 15 traffic sources respectively and obtain similar behaviors. see Fig. 3 and Fig. 4.

Fig. 5 shows that RTS collision can be approximated by Hello loss rates. By increasing the number of nodes or the throughput very highly, we obtain an over estimation of RTS collision probability as explained above. We consider that 20 neighbors is a reasonable number.

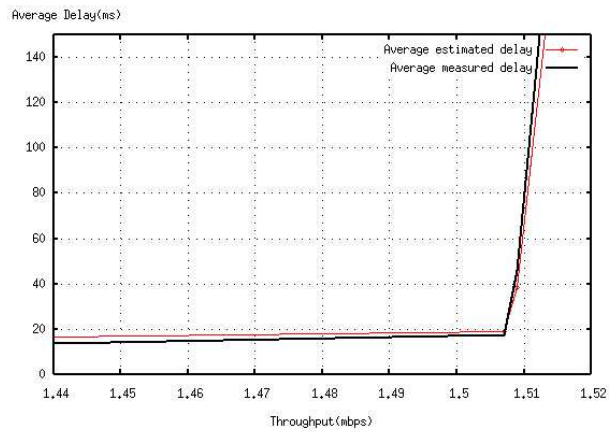

Figure 3: Estimated versus measured delay in 10 nodes network

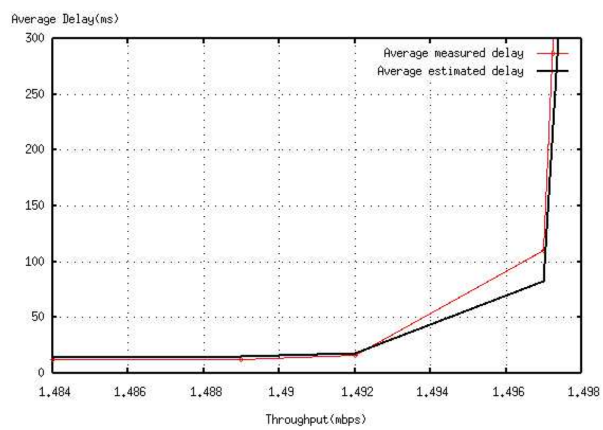

Figure 4: Estimated versus measured delay in 20 nodes network

5 Conclusion

In this report, we present a new approach to evaluating the one-hop delay in a 802.11 ad hoc network using the OLSR protocol. We establish an analytical model and use OLSR statistics to complete it. As we said, Hello loss rate could be refined to be closer to the collision probability. In future work, we will focus on how to obtain the channel occupancy distribution in real environments. The general case of Multi-hop network is being studied.

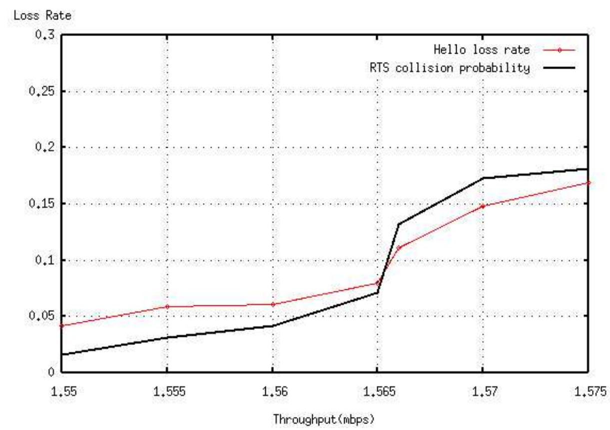

Figure 5: Hello loss versus RTS collision probability in 5 nodes network

Finally, once delay is estimated in an accurate way, it can be optimized by adding admission control mechanism and using delay-oriented routing OLSR.

References

[1] : T. Clausen, P. Jacquet "Optimized Link State Routing Protocol (OLSR)", RFC 3626.

[2] IEEE 802.11 Standard "Wireless Lan Medium Access Control and Physical layer specifications" June 1997.

[3] A. Meraihi Naimi and P. Jacquet: "Delay oriented OLSR" IEEE ICTTA 2004. Syria.

[4] O. Tickoo and B. Sikdar "Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks " IEEE INFOCOM. 2004. Hong Kong.

[5] Bianchi "Perfomance Analysis of the IEEE802.11 Distributed coordination function" IEEE journal on selected areas in Communcations, vol 18, no 3, March 2000.

[6] Y Tay, Chua "Capacity Analysis for the IEEE802.1 Mac Proctocol". Wireless networks, vol. 7, no 2, pp 159-171, March 2001.

[7] H. Zhai, Y. Fang "performance of Wireless Lan Based on IEEE 802.11 Mac Protocol" IEEE PIMRC 2003.

[8] K. Fall, K. Varadhan "The ns manual" http://www.isi.edu/nsnam/doc/ns_doc.pdf. 2002.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399