Signature and Database Exchange for Wireless OSPF Interfaces

Thomas Clausen, Philippe Jacquet, Emmanuel Baccelli

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Signature and Database Exchange for Wireless OSPF Interfaces

Thomas Clausen — Philippe Jacquet — Emmanuel Baccelli

N° 5096

Janvier 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5096--FR+ENG

Signature and Database Exchange for Wireless OSPF Interfaces

Thomas Clausen , Philippe Jacquet , Emmanuel Baccelli

Thème 1 — Réseaux et systèmes
Projets HIPERCOM

Rapport de recherche n° 5096 — Janvier 2004 — 14 pages

Abstract: In this paper, we specify a mechanism for link-state database exchanges in wireless ad-hoc networks. The mechanism is taylored for ad-hoc networks employing the wireless OSPF interface extension specification [1], however is suitable for any proactive link-state routing protocol.

The database exchange mechanism is specified with the following applications in mind:

- reliable diffusion of link-state information, replacing OSPF acknowledgements with a mechanism, suitable for mobile wireless networks;
- reduced overhead for performing OSPF style database exchanges in a mobile wireless network;
- reduced initialization time when new node(s) are emerging in the network;
- reduced overhead and reduced convergence time when two (or more) WOSPF adhoc network clouds merge.

Key-words: ad hoc, wireless, networks, OSPF, database exchange, routing, algorithm

Signature et Echange de Base de Données pour les Interfaces OSPF Sans Fil

Résumé : Dans ce papier nous spécifions un mécanisme d'échange de bases de données portant sur l'état des liens dans un réseau mobile ad hoc. Ce mécanisme est spécifié pour des noeuds mobiles utilisant des interfaces le protocole Wireless OSPF [1], mais peut egalement être adapté a n'importe quel autre protocole proactif basé sur l'état des liens dans le réseau.

Ce mécanisme est spécifié en vue des applications suivantes:

- la diffusion fiable d'information sur l'état des liensdans le réseau, et ceci en remplacement des acquittements du protocole OSPF classique, grâce à un mécanisme adapté à l'environnement sans fil;
- la réduction du coût en bande passante de l'exécution pour un échange de base de données dans un réseau mobile sans fil;
- la diminution sensible des temps d'initialisation quand un noeud apparaît dans le réseau;
- la diminution sensible des durées de convergence lors de la fusion de deux réseaux ad hoc indépendants.

Mots-clés : réseaux, ad hoc, sans-fil, OSPF, échange de base de données, algorithmes, routage

1 Introduction

This document discusses a mechanism providing database exchange and reliable synchronization for wireless OSPF interfaces as defined by draft-spagnolo-manet-ospf-wireless-interface-00 [1].

The basic mechanism is to employ exchange of compact "signatures" between pairs of nodes, in order to detect differences in the nodes' link state databases. When a discrepancy is detected, the required bits of information are then identified and exchanged. The purpose hereof is to provide the nodes with a consistent view of the network topology.

In OSPF [2], the goal of maintaining a consistent view of the network topology in all nodes is achieved through two independent mechanisms: (i) reliable transport of LSA messages and (ii) database exchange, in which a router synchronizes its link-state database with one other router in the network.

OSPF's reliable transport of LSA messages employs positive acknowledgments (ACK) on delivery, with retransmissions if the acknowledgment is missing. I.e. an ACK is a retransmission repressing message. In relatively static point-to-point like network topologies (such as is typically the case with fixed wired networks), ACKs and retransmissions occur over a single link in the network. More importantly, an ACK transmitted by the recipient of an LSA message will be received by a node which is directly able to interpret the ACK message. I.e., the recipient of an ACK will be the node which sent the LSA to which the ACK corresponds.

In the context of wireless OSPF interfaces as defined in [1], the conditions are different: nodes are assumed to be mobile, with network topology changing relatively rapidly as a consequence, and interfaces are broadcast by nature. Hence any transmission (ACK or retransmission) will, at best, interfere with all the neighbors of the node originating the transmission. Therefore, an ACK, which can be correctly interpreted only by the node which sent the LSA to which the ACK corresponds, will still be received by (and interfere with) all the nodes in the neighborhood. Therefore if, due to node mobility or fading radio links, a node does not receive an expected ACK, unnecessary retransmissions will occur, consuming precious bandwidth. Or, in other words,employing reliable topology information diffusion through ACK's imposes the assumption that the network conditions are such that an ACK that is sent can be received by the intended node. This assumption does not hold for OSPF wireless interfaces.

OSPF database exchanges intend to synchronize the link-state database between routers in the network. In OSPF, database description packets are exchanged between two nodes through one node (the master) polling an other node (the slave), which responds. Both polls and responses have the form of database description packets containing a set of complete LSA headers, describing (a partial set of) the respective link-state databases of each of the two nodes. These database description packets are used by the nodes to compare

their link-state databases. If any of the two nodes involved in the exchange detects it has out-of-date or missing information, it issues link-state request packets to request the pieces of information from the other node, which would update its link-state database.

In the context of wireless OSPF interfaces as defined in [1], wireless broadcast interfaces are assumed, as well as a high degree of network topology dynamics. This implies that inconsistencies between the link-state databases of different nodes in the network can be assumed to be occurring more frequently, and that the changes in the topology happen at a much quicker pace than on wired networks. Moreover, the broadcast nature of the network interfaces implies that the bandwidth in a region is shared among the nodes in that region. In terms of database exchange, this implies that this operation may be more frequently employed, with much less time to complete before the topology changes again. And such, over a transmission media with typically a lot less available bandwidth per node pair engaged in the database exchange.

The goal of both the reliable transmission of LSA messages and the database exchange mechanism is, effectively, to ensure that the nodes in the network have the same updated information about the network topology recorded in their respective link-state databases. The mechanism proposed in this document aims at addressing this, in a way which is adapted to link state routing on ad hoc networks in general. However, in the present document it is specified for wireless OSPF interfaces in particular. A signature exchange mechanism is proposed, providing a very compact way for the nodes to communicate the state of their link-state database. Upon receiving of such a signature, a node can quickly detect if a discrepancy is present and, furthermore, determine which information must be exchanged in order to alleviate the discrepancy.

The mechanism described in this document is somewhat inspired by the one employed by IS-IS [3]. In IS-IS, packets which list the most recent sequence number of one or more LSAs (so called Sequence Numbers packets) are used to ensure that neighboring nodes agree on what is the most recent link state information from each other node. I.e., rather than transmitting complete LSA headers (as in OSPF), ISIS employs a more compact representation for database description messages. Additionally, Sequence Numbers packets accomplish a function similar to conventional acknowledgment packets. Sequence Numbers packets also allow more efficient operation, in the sense that they may act as a request for information, i.e., a complete Sequence Number packet containing the most recent sequence number of all the LSAs in the database may be used to ensure synchronization of the database between adjacent routers either periodically, or when a link first comes up, much like the database exchange mechanism.

2 Signature Exchange Protocol

This section details the operation of signature exchange, outlined above. The signature exchange protocol serves the purpose of allowing nodes to detect discrepancies between their respective link-state databases. Correcting such discrepancies once detected, is detailed in section 3.

Section 2.1 gives an abstract definition of the link-state dabase signatures employed. Section 2.2 details how signature exchange is conducted, and section 2.3 and section 2.4 outlines how signatures are generated and checked, respectively.

2.1 Abstract Definition of Link-State Database Signatures

A signature message is a tuple:

signature message = (age interval, key, prefix signature)

with a each prefix signature from the set being defined as:

prefix signature = (prefix, sign(prefix))

The signature for a prefix is constructed thus:

sign(prefix) = (primary partial signature, #LSA,
secondary partial signature,
timed partial signature, timed #LSA)

A primary partial signature for a prefix is computed as a sum over all LSAs in a nodes link-state database where the prefix matches the advertising router of the LSA:

primary partial signature = $\sum_{prefixes}$ (CRC(LSA-identifier))

With $\sum_{prefixes}$ denoting the sum over prefixes matching the advertising router of the LSA. The secondary partial signature for a prefix is computed as a sum over all LSAs in a nodes link-state database, where the prefix matches the advertising router of the LSA:

secondary partial signature = $\sum_{prefixes}$ (CRC(LSA-identifier)) $\cdot$ key

With $\sum_{prefixes}$ denoting the sum over prefixes matching the advertising router of the LSA. The timed partial signature for a prefix and an age interval is computed as the sum over LSAs in a nodes link-state database where:

- the prefix matches the advertising router of the LSA,
- the age falls within the age interval of the advertisement.

It has the following expression:

$$\text{timed partial signature} \quad = \quad \sum_{prefixes,time} (\text{CRC(LSA-identifier)})$$

With $sum\_prefixes, time$ denoting the sum over prefixes matching the advertising router of the LSA and where the age falls within the age interval of the advertisement. The LSA identifier is the string, obtained through concatenating the following fields from the LSA header:

- LS type
- LS ID
- Advertising router
- LSA sequence number

2.2 Signature Exchange

Signatures are exchanged between nodes in two forms: informational signatures, which are broadcast periodically to all neighbor nodes, and database exchange signatures, employed when a node requests a database exchange with one of its neighbors.

2.2.1 Informational Signatures

Each node periodically broadcasts informational (info) signature, as well as receives signatures from its neighbor nodes. This exchange allows nodes to detect any discrepancies between their respective link-state databases. Section 2.3.1 details how info signatures are generated; section 2.4 details how signatures are employed to detect link-state database discrepancies.

2.2.2 Database Exchange Signature

Contrary to the informative signatures, Database Exchange (dbx) signatures are directed towards a single neighbor only. The purpose of emitting a dbx signature is for a node to initiate an exchange of database information with a specific neighbor node.

When a node detects a discrepancy between its own link-state database and the link-state database of one (or more) of its neighbors, a database exchange is desired to eliminate that discrepancy. The node, detecting the discrepancy, generates a dbx signature, effectively requesting the database exchange to take place. In OSPF terms, the node requesting the database exchange is the "master" of that exchange. The dbx signature is transmitted with the destination address of one node among the discrepant neighbors. In OSPF-terms, that neighbor node would be the "slave" in the database exchange. If possible, the slave should be an MPR for the selecting node. The node build a dbx message signature, based on the

information acquired from the info signature exchange. Section 2.3 details how dbx signatures are generated. Section 2.4 details how signatures are checked by a node, in order to detect link-state database discrepancies. Section 3 details how the actual database exchange is performed.

2.3 Signature message generation

This section details how signature messages are generated. There are two types of signature messages: (i) informative signature messages, and (ii) database exchange signature messages.

2.3.1 Info Signatures

An informative signature message (also called info signature) can be sent for periodical mutual check and describes the complete link state database of the node that sends it. Namely, if no information is given here about a given prefix, a neighbor can implicitly understand that the node has no corresponding LSA in its database.

The set of prefix signatures in an informative signature message can be generated with the following algorithm (Tunstall Splitting), where the length L of the info signature (the number of prefix signatures in the message) can be chosen at will.

We define the weight, and the timed weight, of a given prefix as the functions:

Weight(prefix) = number of LSAs whose originator matches the prefix.

Timed Weight(prefix) = number of LSAs whose originator matches the prefix and whose age falls inside the age interval.

Then, starting with the set of prefix signatures equal to (0,signature(0)), recursively do the following.

As long as:

|set of prefix signatures| < L

1. Find in the set of prefix signatures the prefix with largest timed weight, let it be called mprefix.
2. Replace the single (mprefix,signature(mprefix)) by the pair (mprefix0,signature(mprefix0)), (mprefix1,signature(mprefix1)).
3. If one of the expanded prefix of mprefix has weight equal to 0, then remove the corresponding tuple.

2.3.2 dbx Signatures

When a node realizes there are desprepencies between his database and the signature sent by neighbors, it sends a database exchange signature message (also called dbx signature) to trigger the exchange of descrepant LSAs with one of these neighbors, preferably an MPR, which is designated in the destination field.

The set of prefix signatures in a database exchange signature message can be generated with the following algorithm, where the length L of the dbx signature (the number of prefix signatures in the message) can be chosen at will.

Start with the same set of prefix signatures as one of the received info signature where the descrepencies were noticed.

Remove from that set all the prefix signatures such that signature(prefix) is not descrepant (with the LSA database). Use the same age interval and key used in the received info signature. Then recursively do the following.

As long as:

$|\text{set of prefix signatures}| < \text{L}$

1. Find in the set of prefix signatures the prefix with largest timed weight, let it be called mprefix.
2. Replace the single (mprefix,signature(mprefix)) by the pair (mprefix0,signature(mprefix0)), (mprefix1,signature(mprefix1)).
3. Notice that contrary to info signature messages, the prefixes with zero weight are not removed here, since the signature is not complete, i.e. the signature might not describe the whole database. Therefore a prefix with empty weight may be an indication of missing LSAs.

2.4 Checking Signatures

Upon receiving a signature message from a neighbor, a node can check its local LSA database and determine if it differs with the neighbor's. For this purpose, it computes its own prefix signatures locally using the same prefixes, time interval and key specified in the received signature message. A prefix signature differs with the local prefix signature when any of the following conditions occurs:

1. both the number of LSAs and the timed number of LSAs differ;
2. both the timed partial signatures and the (primary partial signature, secondary partial signature) tuples differ.

The use of a secondary signature based on a random key is a way to cope with the unfrequent but still possible cases when the primary signatures agree although the databases differ. In this case, it can be assumed that using a random key renders the probability that both primary and secondary signatures agree while databases are different, to be very small.

3 Database Exchange

When a node receives a dbx signature with its own ID in the destination field, the node has been identified as the slave for a database exchange. The task is, then, to ensure that information is exchanged to remove the discrepancies between the link-state databases of the master and the slave.

Thus, the slave must identify which LSA messages it must retransmit, in order to bring the information in the master up-to-date. The slave must then proceed to rebroadcast those LSA messages.

More precisely, the slave rebroadcasts the LSA messages which match the following criteria:

- the age belongs to the age interval indicated in the dbx signature, AND
- the prefix corresponds to a signed prefix in the dbx signature, where the signature generated by the master differs from the signature as calculated within the slave for the same segment of the link-state database.

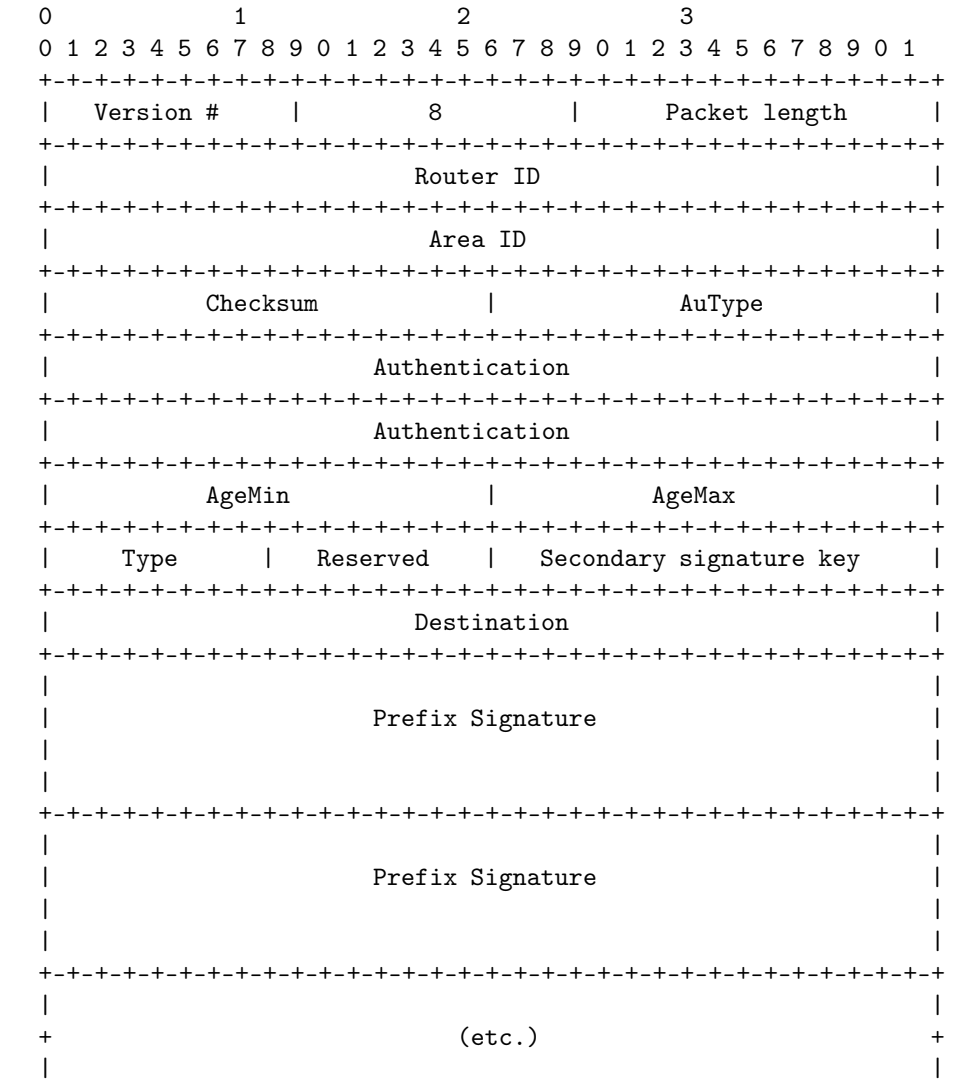
When a node is triggered to perform a database exchange it generates a new LSF with TTL equal to 1 (one hop only) and fills it with the update LSAs. These LSAs must indicate the age featured at the moment in the database, from which they are taken.

Optionally, the host can use a new type of LSF (denoted an LSF-D) which, contrary to the one hop LSF described above, is retransmitted as a normal LSF making use of MPRs. An LSF-D is transmitted with TTL equal to infinity. Upon receiving of such a packet, successive nodes remove from the LSF-D the LSAs already present in their database before retransmitting the LSF-D. If the LSF-D is empty after such a processing, a node will simply not retransmit the LSF-D. The use of LSF-D packets is more efficient for fast wide-area database updates in case of merging of two independent wireless networks.

4 Packet Formats

Info and dbx signatures share the same packet format, detailed in this section.

4.1 Signature package format



Version #, Packet length, Router ID, Area ID, Checksum, AuType and Authentication fields are the OSPF control packet header as described in [OSPF]

AgeMin, AgeMax

AgeMin and AgeMax defines the age interval [AgeMin,AgeMax], used for computing the timed partial signatures in the prefix signatures as described in section 2.1.

Type

Specifies if the signature is an info or a dbx signature, according to the following:

Value	Type
1	info (informative)
2	dbx (database exchange)

Reserved

Must be set to "00000000" for compliance with this specification.

Secondary signature key

The key of the secondary signature is a random number of 32 bits. Used for computing the secondary partial signature as described in section 2.1.

Destination

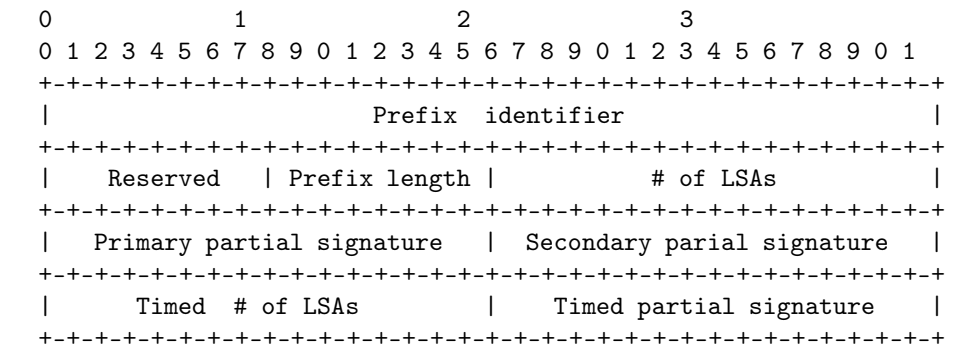
If the signature is of type = 2, then this field contains the address of the slave, with which a database exchange is requested.

If the signature is of type = 1, then this field must be zero'ed

Prefix signature

The set of prefixes signatures contains the sub-signatures for different parts of the link-state database. The layout of the prefix signatures is detailed in section 4.2.

4.2 Prefix Signature Format



Prefix identifier and Prefix length

Indicates the length of the prefix for the part of the link-state database, as well as the exact prefix.

# of LSAs

The number of LSAs in the emitting nodes link-state database, matching by the prefix identifier and prefix legth.

Primary partial signature

The arithmetic sum of the CRCs of each string made of the concatenation of sequence number and LSA-originator ID fields of the tuples (LSA-originator ID, LSAsequence-number, LSA-age) from the emitting nodes link-state database such that the LSA-originator ID and prefix ID has same prefix of length prefix-length.

Secondary partial signature

The arithmetic sum of the XOR between the secondary signature key and each of the CRCs of each string made of the concatenation of sequence number and LSA-originator ID fields of the tuples (LSA-originator ID, LSAsequence-number, LSA-age) from the emitting nodes link-state database such that the LSA-originator ID and prefix ID has same prefix of length prefix-length.

Timed # of LSAs

The number of LSAs in the emitting nodes link-state database, matching by the prefix identifier and prefix legth and satisfying the condition that the LSA age is between AgeMin and AgeMax.

Timed partial signature

The arithmetic sum of the CRCs of each string made of the concatenation of sequence number and LSA-originator ID fields of the tuples (LSA-originator-ID,LSA sequence-number, LSA-age) from the emitting nodes link-state database such that:

- Prefix ID and LSA-originator ID has same prefix of length prefix-length
- LSA-age is between AgeMin and AgeMax.

5 Applicability of the signature and database exchange mechanism

This section outlines the applicability of the specified mechanisms in a set of common scenarios.

5.1 Emerging node

When a new node emerges in an existing network, the initialization time for that node is the time until it has acquired link-state information, allowing it to participate fully in the network. Ordinarily, this time is determined solely by the frequency of control traffic transmissions. In order to reduce the initialization time, the database exchange mechanisms can be employed as soon as the node has established a relationship with one neighbor node already initialized. This emerging node will select a neighbor as slave and transmit a dbx signature of the form ([age min, age max],(*,signature(*)), "*" implying an empty prefix. The slave will respond by, effectively, offering its entire link-state database to the master. In particular in situations where the some LSAs are not transmitted frequently (outside LSAs would be an example of such), this mechanism may drastically reduce the initialization time of new nodes in the network.

5.2 Merging Wireless Clouds

Two disjoint sets of nodes, employing [1] as their routing protocol, may at some point merge or join – i.e. that a direct (radio) link is established. Prior to the merger, the respective clouds are "stable", periodically broadcasting consistent info signatures within their respective networks. At the point of merger, at least two nodes, one from each network,

will be able to establish a direct link and exchange control traffic. The combined network is now in an unstable state, with great discrepancies between the link-state databases of the nodes in the formerly two networks. Employing signature and database exchanges through the LSF-D mechanism, the convergence time until a new stable state is achieved can be kept at a minimum.

5.3 Acknowledgments

If a node wants a specific LSA to be reliably transmitted to its neighbor, the db signature mechanism can be employed outside of general periodic signature consistency check. The node transmitting the LSA message broadcasts an info signature, containing the full LSA-originator ID as signed prefix and a very narrow age interval, centered on the age of the LSA which is to be reliably transmitted. A neighbor which does not have the LSA in its database will therefore automatically trigger a database exchange concerning this LSA and send a dbx signature containing the LSA-originator ID signed with an empty signature. The receiving of such a dbx signature will trigger the first node to retransmit the LSA right away with a new LSF to ensure that the LSA does get through.

References

[1] J. Ahrenholz, T. Henderson, P. Spagnolo, P. Jacquet, E. Baccelli, T. Clausen. OSPFv2 Wireless Interface Type. draft-spagnolo-manet-ospf-wireless-interface-00.txt, Internet Engineering Task Force, November 2003.

[2] J. Moy, "OSPF version 2," RFC 2328, http://ietf.org/rfc/rfc2328.txt, 1998.

[3] D. Oran, "OSI IS-IS Intra-domain Routing Protocol," RFC 1142, http://ietf.org/rfc/rfc1142.txt, 1990.

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

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY 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 - 38334 Montbonnot Saint-Ismier (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