

OSPF DATABASE EXCHANGE AND RELIABLE SYNCHRONIZATION IN MOBILE AD HOC NETWORKS

Emmanuel Baccelli
INRIA Rocquencourt
Hipercom Project
email: Emmanuel.Baccelli@inria.fr

Thomas Heide Clausen
INRIA Rocquencourt
Hipercom Project
email: T.Clausen@computer.org

Philippe Jacquet
INRIA Rocquencourt
Hipercom Project
email: Philippe.Jacquet@inria.fr

ABSTRACT

In this paper, we expose the need for an alternate mechanism replacing usual database exchange and reliable synchronization in OSPF for mobile ad hoc environments. A mechanism for link-state database exchanges in wireless ad-hoc networks is proposed, aiming at ‘completing the adaptation of OSPF for ad hoc networks started in [1]. This adapted mechanism is specified with the following applications in mind: (i) Reliable diffusion of link-state information replacing OSPF acknowledgements with a mechanism suitable for mobile wireless networks; (ii) Reduced overhead for performing OSPF style database exchanges in a mobile wireless network; (iii) Reduced initialization time when new nodes are emerging in the network; (iv) Reduced overhead and reduced convergence time when several wireless OSPF ad hoc network clouds merge. **KEY WORDS: mobile, ad hoc, network, OSPF, database, synchronization.**

1 Introduction

The possible use of OSPF as a routing protocol in wireless ad hoc networks has lately been the subject of several different efforts. OLSR [4], a link-state protocol developed by the IETF specifically for routing in wireless ad-hoc networks, is in its essential functioning very close to that of OSPF, yet is without several key features of OSPF – notably a close integration between wired and wireless ad-hoc routing.

There is indeed a need for a generic wired/wireless IP routing solution. Due to its widespread use on wired networks, as well as its likeness to OLSR, OSPF seems like a designated candidate. However, OSPF in its basic form is not at all tailored for mobile wireless environments and features several problems when run in these [6] [7].

A solution for making OSPF operate efficiently on ad hoc networks is proposed in [1], where a new type of OSPF interface is specifically defined for manet interfaces. However [1] proposes a partial adaptation of OSPF for wireless ad-hoc networks: adjacencies are not formed on wireless ad-hoc network interfaces, which implies that the usual OSPF database exchange and reliable synchronization mechanisms are not in action on these interfaces. Rather, [1] features simple periodic updating and sending of LSAs

in LSFs with a very short period, the same way OLSR diffusion of link-state information (through TC messages) is conducted. The idea being that since the topology of the network is thought to be changing frequently, LSA’s are transmitted periodically and frequently to reflect these changes. Consequently, loss of a single LSA is relatively unimportant since the information contained within the message will be repeated shortly. This approach may not work well if LSA periods are not roughly homogeneous and short: in a mixed wired/wireless network, the LSAs generated by usual wired nodes will have long periods (up to 1hr) while LSAs generated by wireless nodes will typically have a period of (often much) less than a minute. In this case, of course, the short period argument fails, at least for the LSAs with a long period, and there is a definite need for a mechanism to device mechanisms for conducting the usual OSPF database exchange and reliable synchronization in a wireless ad-hoc network.

This paper proposes a technique for conducting efficient database synchronization that is adapted to link state routing on ad hoc networks in general, and to wireless OSPF interfaces as defined in [1] in particular.

The following section 2 will state the problem with usual OSPF database exchange and reliable synchronization in a mobile ad-hoc environment. Section 3 proposes a solution: the database signature exchange mechanism, and describes its details in the context of [1]. Section 4 presents a performance evaluation of this technique, section 5 discusses its applicability, and section 6 concludes this paper.

2 OSPF Mechanisms and Issues on Ad Hoc Networks

A link state routing protocol’s essential function is to maintain, in each node forming the network, a consistent view of the global topology (the so-called link state database). In OSPF [2], this consistency 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.

2.1 Reliable Transmission

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.

However, in the context of OSPF over mobile wireless ad hoc interfaces 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. 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 on wireless interfaces.

2.2 Database Exchange

OSPF database exchanges are intended 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 OSPF over mobile ad hoc radio interfaces, 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 database exchange operation may be more frequently employed, with much less time to complete before the topology changes again. This, over a transmission media with typically a lot less available bandwidth per node pair engaged in the database exchange.

3 Database Signature Exchange

This section proposes a mechanism providing database exchange and reliable synchronization for wireless OSPF interfaces as defined in [1]. The basic principle of database signature exchange is to employ an exchange of compact "signatures" (hashing of the link state database) between neighbor nodes, in order to detect differences in the nodes' link state databases. When a discrepancy is detected, the bits of information required to synchronize the link state databases of the involved nodes are then identified and exchanged. The purpose of the exchange is to provide the nodes with a consistent view of the network topology – the task is doing so in an efficient way.

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 synchronization of the database between adjacent routers either periodically, or when a link first comes up, much like the database exchange mechanism of OSPF.

The database signature exchange, proposed in this paper, differs from the mechanism employed in IS-IS with the use of age. The signature messages proposed can make use of boundaries on the age of the LSAs which make the basis for signature computation. For example, it may be considered a waste of resources to check for databases consistency for LSAs issued from within wireless environments: LSAs from wireless nodes are transmitted frequently and periodically, thus information describing these nodes is frequently updated and thereby of an age smaller than a threshold, roughly close to the LSF period. With this perspective, if one wants to limit the scope of the database synchronization exchanges to the LSAs which are less frequently updated (i.e. LSAs issued from a wired environment), it suffices to set the LSA age floor limit above the LSF period and the LSA age ceiling to MaxAge in the signature messages.

The following subsections detail the operation of the

database signature exchange, outlined above. The database signature exchange protocol allows nodes to detect discrepancies between their respective link-state databases. Correcting such discrepancies once detected, is detailed in section 3.5. Section 3.1 gives an abstract definition of the link-state database signatures employed. Section 3.2 details how signature exchange is conducted. Section 3.3 and section 3.4, respectively, outline how signatures are generated and checked.

3.1 Mathematical Definition of Link State Database Signatures

We define a signature message as a tuple of the following form:

$$\text{Signature Message} = (\text{Age Interval}, \text{Key}, \text{Prefix Signature}),$$

A signature features a set of prefix signatures, each one of them being of the form:

$$\text{Prefix Signature} = (\text{Prefix}, \text{Sign}(\text{Prefix})).$$

Each $\text{Sign}(\text{Prefix})$ results from hashing functions computed on the piece of the link state database matching with the specified prefix, and represents this part of the database in the signature message. More specifically, each $\text{Sign}(\text{Prefix})$ has the following structure:

$$\text{Sign}(\text{Prefix}) = (\text{Primary Partial Signature}, \text{Secondary Partial Signature}, \text{Timed Partial Signature}, \#LSA, \text{Timed } \#LSA).$$

A primary partial signature (or PPS) 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:

$$\text{PPS} = \sum_{\text{prefixes}} (\text{Hash}(\text{LSA} - \text{identifier})),$$

with \sum_{prefixes} denoting the sum over prefixes matching the advertising router of the LSA. The secondary partial signature (or SPS) 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:

$$\text{SPS} = \sum_{\text{prefixes}} (\text{Hash}(\text{LSA} - \text{identifier})) \cdot \text{key},$$

with \sum_{prefixes} denoting the sum over prefixes matching the advertising router of the LSA. The timed partial signature (or TPS) for a prefix and an age interval is computed over LSAs in a nodes link-state database where: (i) the prefix matches the advertising router of the LSA, (ii) the age falls within the age interval of the advertisement, and has the following expression:

$$\text{TPS} = \sum_{\text{prefixes,time}} (\text{Hash}(\text{LSA} - \text{identifier})),$$

with $\sum_{\text{prefixes,time}}$ 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.

3.2 Signature Exchange

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

3.2.1 Informational Signature Exchange

Each node periodically broadcasts informational (info) signatures, as well as receives signatures from its neighbor nodes. This exchange allows nodes to detect any discrepancies between their respective link-state databases. Section 3.3 details how info signatures are generated; section 3.4 details how signatures are employed to detect link-state database discrepancies.

3.2.2 Database Signature Exchange

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. The node builds a dbx message signature, based on the information acquired from the info signature exchange. Section 3.3 details how dbx signatures are generated. Section 3.4 details how signatures are checked by a node, in order to detect link-state database discrepancies. Section 3.5 details how the actual database exchange is performed.

3.3 Signature message generation

This section details how signature messages are generated. As previously mentioned, there are two types of signature messages: (i) informative signature messages, and (ii) database exchange signature messages. Their actual packet format is described in appendix A.

3.3.1 Info Signatures

An informative signature message (also called info signature) can be sent to periodically allow neighbor nodes to check their link state databases against the link state database of the sender of the informative signature. Thus, an informative signature describes the complete link state database of the node that sends it. Absence of information in a signature indicates absence of information in the sending node's link state database – or, in other words, if no information is given within an informational signature about a specific prefix, it is implicitly to be understood that the sending node has received no LSAs corresponding to that prefix.

The set of prefix signatures in an informative signature message can be generated with the following splitting algorithm, 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 of a given prefix as the function:

$$\text{Weight}(\text{prefix}) = \text{number of LSAs whose originator matches the prefix.}$$

And similarly, the timed weight as the function:

$$\text{Timed Weight}(\text{prefix}) = \text{number of LSAs whose originator matches the prefix and whose age falls inside the age interval.}$$

Then, starting with the set of prefix signatures equal to $(0, \text{signature}(0))$, recursively do the following. As long as: $|\text{set of prefix signatures}| < L$

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

3.3.2 Dbx Signatures

When a node, through receipt of informative signatures, realizes there are discrepancies between its own link state

database and that of one of its neighbors, it sends a database exchange signature message (also called dbx signature) to trigger the exchange of discrepant LSAs with one of these neighbors. Care must be taken when selecting with which neighbor(s) a database exchange is conducted – the ideal is to pick a node which has the “most complete” link-state database and which at the same time is going to remain a neighbor for a sufficient period of time. For [1], database exchanges are to be conducted in preference with nodes selected as MPR. The node with which a database exchange is to be conducted is designated in the destination field of the signature message.

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 discrepancies were noticed. Then remove from that set all the prefix signatures such that $\text{signature}(\text{prefix})$ is not discrepant (with the LSA database). Use the same age interval and key used in the received info signature. Then use the recursive algorithm described in section 3.3.1, skipping step 3. Indeed, 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.

3.4 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 database. 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: either (i) both the number of LSAs and the timed number of LSAs differ; or (ii) 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, situations 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.

3.5 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: (i) the age belongs to the age interval indicated in the dbx signature, and (ii) 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 Performance Evaluation of the Database Signature Exchange Mechanism

This section evaluates the performance of the Database Signature Exchange mechanism compared to OSPF's full database exchange. Figure 1 shows the retrieval cost implied by each method, in the case of a single record mismatch: the less the databases are different, the more Database Signatures perform better compared to OSPF's full database exchange. While OSPF's DB Exchange cost grows linearly with the size of the database, the cost of the Signature Exchange remains very low and therefore achieves a significant improvement.

5 Applicability of the Database Signature Exchange Mechanism

This section outlines further applicability scenarios for the specified mechanism.

- **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

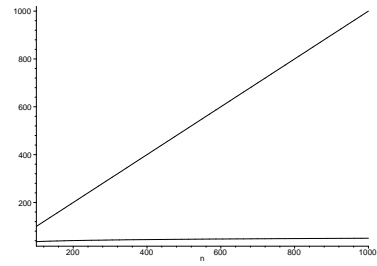


Figure 1. A signature retrieval cost (bottom) compared with full database retrieval cost (top) with a single record mismatch, plotted as a function of the database size.

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.

- **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 transmitting 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.
- **Reliable Flooding** - 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.

6 Conclusion

In this paper, we have exposed the limitations on mobile ad hoc networks of some essential mechanism in OSPFv2: database exchange and reliable synchronization. We have proposed another mechanism adapted to such an environment, presented for instance in this document as completing the work done in [1], in order to further adapt OSPF to mobile ad hoc networks.

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.
- [4] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol," RFC 3626, <http://ietf.org/rfc/rfc3626.txt>, 2003.
- [5] S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, <http://ietf.org/rfc/rfc2501.txt>, 1999.
- [6] F. Baker, M. Chandra, R. White, J. Macker, T. Henderson, E. Baccelli, "MANET OSPF Problem Statement," Internet-Draft: draft-baker-manet-ospf-problem-statement-00.txt, Oct. 2003.
- [7] C. Adjih, E. Baccelli, P. Jacquet, "Link State Routing in Wireless Ad Hoc Networks," MILCOM 2003 Proceedings.

A Packet Formats

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

A.1 Signature Packet Format

Version #, Packet length, Router ID, Area ID, Checksum, AuType and Authentication fields are the OSPF control packet header as described in [2].

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 3.3.

Type

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

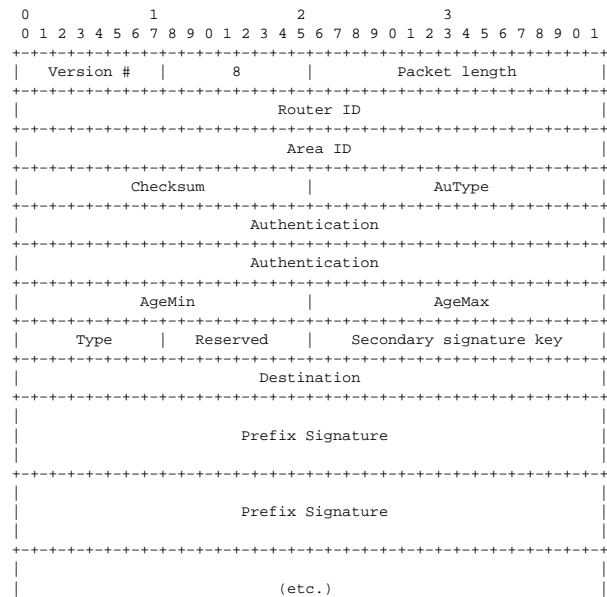
| Value | Type |
|-------|-------------------------|
| 1 | info (informative) |
| 2 | dbx (database exchange) |

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 3.1.



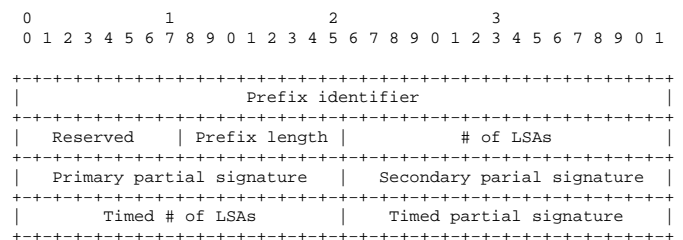
Destination

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 3.3.

A.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 length.

Primary partial signature

The arithmetic sum of the hashing 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 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 hashing 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 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 length and satisfying the condition that the LSA age is between AgeMin and AgeMax.

Timed partial signature

The arithmetic sum of the hashing 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.