



HAL
open science

Database Exchanges for Ad-hoc Networks Using Proactive Link State Protocols

Emmanuel Baccelli, Philippe Jacquet, Thomas Heide Clausen

► **To cite this version:**

Emmanuel Baccelli, Philippe Jacquet, Thomas Heide Clausen. Database Exchanges for Ad-hoc Networks Using Proactive Link State Protocols. Performance Modelling and Analysis of Heterogeneous Networks, River Publishers, pp.93 - 111, 2009. hal-00651457

HAL Id: hal-00651457

<https://hal.science/hal-00651457>

Submitted on 13 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Database Exchanges for Ad-hoc Networks Using Proactive Link State Protocols

E. Baccelli, P. Jacquet¹, T. Clausen²

December 13, 2011

¹INRIA, France. {Emmanuel.Baccelli, Philippe.Jacquet}@inria.fr

²Ecole Polytechnique, France. T.Clausen@computer.org

Abstract

The OSPF routing protocol is currently the predominant IGP in use on the fixed Internet of today. This routing protocol scales “world wide”, under the assumptions of links being relatively stable, network density being rather low (relatively few adjacencies per router) and mobility being present at the edges of the networks only. Recently, work has begun towards extending the domain of OSPF to also include ad-hoc networks – i.e. dense networks, in which links are short-lived and most nodes are mobile.

In this paper, we focus on the convergence of the Internet and ad-hoc networks, through extensions to the OSPF routing protocol. Based on WOSPF, a merger of the ad-hoc routing protocol OLSR and OSPF, we examine the feature of OSPF database exchange and reliable synchronisation in the context of ad-hoc networking. We find that the mechanisms, in the form present in OSPF, are not suitable for the ad-hoc domain. We propose an alternative mechanism for link-state database exchanges in wireless ad-hoc networks, aiming at furthering an adaptation of OSPF to be useful also on ad-hoc networks, and evaluate our alternative against the mechanism found in OSPF.

Our proposed 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 initialisation 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.

Chapter 1

-

1.1 Introduction

Wireless ad-hoc networks are characterised by being networks of autonomous and mobile nodes, communicating over a wireless medium whereby they form an arbitrary, dynamic and random graph of wireless links. When the network size grows to the point where direct links do no longer exist between all node-pairs, ensuring connectivity in such a network becomes the task of *routing*. Routing in wireless ad-hoc networks brings a host of challenges not present in traditional wired networks, including “hidden nodes”, low and commonly shared bandwidth, limited resources in the nodes (processing- and battery-power), a high degree of network dynamics etc.

The possible use of OSPF [2] as a routing protocol in such wireless ad hoc networks has lately been the subject of several different efforts. OLSR [4], a link-state protocol developed within 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 the ability to perform routing in a heterogeneous environment such as 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 wireless ad hoc networks is Wireless-OSPF (WOSPF), proposed in [1], where a new type of OSPF interface is specifically defined for manet interfaces. This interface type operates

through employing the ad-hoc network specific optimisations of OLSR (*i.e.* periodic unreliable message transmission and optimised flooding through multipoint relays [8]) while maintaining OSPF messages (*i.e.* Link-State Advertisements - LSA) for diffusing topological information. However [1] proposes only 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 synchronisation mechanisms are not in action on these interfaces.

The idea behind the periodic unreliable flooding of topology information is, that since the topology of the network is thought to be changing frequently, LSAs (in OSPF) and TCs (similar messages in OLSR, called Topology Control messages) are transmitted periodically and frequently to reflect these changes. Consequently, loss of a single LSA or TC is relatively unimportant since the information contained within the message will be repeated shortly. This approach may not work well if LSA or TC periods are not roughly homogeneous and short: in a heterogeneous wired/wireless network, the LSAs generated by usual wired nodes running OSPF will have long periods (up to 1 hour) while LSAs generated by wireless nodes (running WOSPF) 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 synchronisation in a wireless ad-hoc network.

In this paper we propose a mechanism, adapted for the low-bandwidth high-dynamics conditions of wireless ad-hoc networks, for conducting efficient database synchronisation in WOSPF. We qualify the performance of the proposed mechanism and compared to the performance of the original mechanism of OSPF. We furthermore discuss a selection of applicability scenarios for the mechanism, including reliable diffusion of link-state information through, reduced overhead for performing OSPF-style database exchanges in a wireless ad-hoc network, reduced initialisation time when new nodes are emerging in the network and reduced overhead and reduced convergence time when several network clouds merge.

1.1.1 Outline

The remainder of this paper is organised as follows: in section 1.2, a brief description of the usual OSPF database exchange and reliable synchronisation mechanisms is reviewed, and briefly discussed in the context of wireless ad-hoc networks.

Following, section 1.3 describes a mechanism for conducting database exchange and reliable synchronisation, specifically adapted to, and described in the context of, WOSPF [1]. This mechanism respects the fact that not all LSA's carry

information which is long-lived enough to justify the efforts of maintaining consistency, while it still provides an efficient mechanism for allowing nodes to maintain consistency when needed.

Section 1.4 evaluates the performance of the proposed signature exchange mechanism in comparison with the performance of the native OSPF signature exchange mechanism. Following, section 1.5 discusses the applicability of the proposed database exchange mechanism, and section 1.6 concludes this paper.

1.2 Database Exchange in OSPF

The objective of the OSPF routing protocol is to provide, in each node, sufficient topological information about the network to be able to compute (using some metric) a suitable path between any source and destination in the network.

OSPF [2] employs two independent mechanisms for maintaining globally consistent topology information in the node: (i) reliable transport of LSA messages and (ii) database exchanges between pairs of routers.

1.2.1 Reliable Transmission

OSPF employs positive acknowledgements (ACK) on delivery with retransmissions. I.e. an ACK is a retransmission repressing message. In mostly static point-to-point-like network topologies (e.g. 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 wireless ad-hoc networks, the network topology may be assumed to be changing frequently (node mobility). Interfaces are typically wireless (hence subject to fading), of broadcast nature. Any transmission may thus interfere with all the neighbours of the node originating the transmission. An ACK, which can be interpreted by the node which relayed the to the ACK corresponding LSA, will thus be interfering with all the nodes in the neighbourhood. If, due to node mobility or fading radio links, a node does not receive an expected ACK, unnecessary retransmissions will occur, consuming precious bandwidth. In other words, 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 does not hold for a wireless ad-hoc network,

where the network may be substantially more dynamic: nodes may move out of range etc.

1.2.2 Database Exchange

OSPF database exchanges are intended to synchronise the link-state database between routers. In OSPF, database description packets are exchanged between two nodes through one node (the master) polling an other node (the slave). 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 ad-hoc networks, wireless broadcast interfaces and a higher degree of node mobility are typically assumed. Therefore, inconsistencies between the link-state databases of the nodes in the network may occur more frequently, calling for more frequent database exchanges. Moreover, the broadcast nature of the network interfaces implies that the bandwidth in a region is shared among the nodes in that region and thus less bandwidth is available between any pair of nodes to conduct the database exchange.

1.3 Database Signature Exchange

In this section, we propose a mechanism for database exchange and reliable synchronisation, adapted to wireless ad-hoc networks. Specifically, we propose an extension to WOSPF.

The basic idea is to employ an exchange of compact "signatures" (hashing of the link state database) between neighbour nodes, in order to detect differences in the nodes' link state databases. When a discrepancy is detected, the bits of information required to synchronise 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.

Our approach is somewhat inspired by IS-IS [3], in which packets which list the most recent sequence number of one or more LSAs (Sequence Numbers

packets) are used to ensure that neighbouring nodes agree on the most recent link state information. I.e., rather than transmitting complete LSA headers (as in OSPF), a more compact representation for database description messages is employed. Also, Sequence Numbers packets accomplish a function, similar to conventional acknowledgement packets.

The method proposed in this paper differs from the mechanism employed in IS-IS by the use of age. For example, it may be considered a waste of resources to check for databases consistency for LSAs issued from within a very dynamic part of a wireless ad-hoc network (e.g. RFID tags on products in a plant): LSAs from nodes within this domain should be transmitted frequently and periodically, thus information describing these nodes is frequently updated and “with a small age”. LSAs from a less mobile part of the wireless ad-hoc network (e.g. sensors on semi-permanent installations in the plant) might be updated less frequently. Thus consistency of the corresponding entries in the link-state databases should be ensured.

The following subsections outline how database signatures are generated, exchanged, interpreted and used for correcting discrepancies.

1.3.1 Definition of Link State Database Signatures

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

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

A signature features a set of prefix signatures:

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

Each $\textit{Sign}(\textit{Prefix})$ results from hashing functions computed on the piece of the link state database matching the specified prefix, and represents this part of the database in the signature message.

More specifically, each $\textit{Sign}(\textit{Prefix})$ has the following structure:

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

A primary partial signature (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:

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

$\sum_{prefixes}$ denotes the sum over prefixes matching the advertising router of the LSA. The secondary partial signature (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:

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

$\sum_{prefixes}$ denotes 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:

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

and has the following expression:

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

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 LSA header fields:

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

1.3.2 Signature Exchange

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

Informational Signature Exchange – Each node periodically broadcasts informational (info) signatures, as well as receives signatures from its neighbour nodes. This exchange allows nodes to detect any discrepancies between their respective link-state databases. Section 1.3.3 details how info signatures are generated; section 1.3.4 details how signatures are employed to detect link-state database discrepancies.

Database Signature Exchange – Database exchange (dbx) signatures are directed towards a single neighbour only. The purpose of emitting a dbx signature is to initiate an exchange of database information with a specific neighbour node.

When a node detects a discrepancy between its own link-state database and the link-state database of one of its neighbours, a database exchange is desired. The node, detecting the discrepancy, generates a dbx signature, requesting a database exchange to take place. In OSPF terms, the node requesting the database exchange is the "master" while the node selected for receiving the dbx signature is the "slave" of that exchange. The dbx signature is transmitted with the destination address of one node among the discrepant neighbours. The node builds a dbx message signature, based on the information acquired from the info signature exchange.

1.3.3 Signature message generation

This section details how info and dbx signature messages are generated.

Info Signature Generation – An info signature message 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 nodes link state database – 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:

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

And similarly, the timed weight as the function:

Timed Weight(prefix) = # 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 *mprefix*.
2. Replace the single $(\text{mprefix}, \text{signature}(\text{mprefix}))$ by the pair $(\text{mprefix0}, \text{signature}(\text{mprefix0}))$, $(\text{mprefix1}, \text{signature}(\text{mprefix1}))$.
3. If one of the expanded prefix of *mprefix* has weight equal to 0, then remove the corresponding tuple.

Dbx Signature Generation – Dbx signatures serve to trigger an exchange of discrepant LSAs with one neighbour, known to have more up-to-date link-state information – the ideal is to pick the neighbour which has the “most complete” link-state database and which at the same time is going to remain a neighbour for a sufficient period of time. In WOSPF, database exchanges are to be conducted in preference with nodes selected as MPR.

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.

1. Start with the same set of prefix signatures as one of the received info signature where the discrepancies were noticed.
2. 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 above for info signatures, 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.

1.3.4 Checking Signatures

Upon receiving a signature message from a neighbour, a node can check its local LSA database and determine if it differs with the neighbour'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:

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 infrequent, 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.

1.3.5 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:

- 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.

1.4 Performance Evaluation

In this section we compare the performance of database signature exchange protocol with the full database exchange of OSPF. In this first analysis we consider the “cost” of the protocol when two databases differ on a single record. While this is a special case, it gives a good idea of what kind of performance gains we can get.

We denote by n the number of records in the database (typically n can range from a few tens to a thousand) and by Q the number of aggregated signatures contained in a signature message (typically $Q = 10$). Quantity b denotes the maximal size of the portion of database a node will transmit as a whole (*i.e.* without signature exchanges). To simplify we assume that a signature and a record exchange yields the same cost. Let this cost be the unit.

1.4.1 Retrieving a single mismatch

Let D_n being the average cost of database retrieval when the mismatch occurs on a random record among n . Let S_n be the average retrieval costs summed on all possible location of the mismatch in the database. Typically $S_n = nD_n$.

Theorem 1 *The average recovery cost of a single mismatch is:*

$$\begin{aligned} D_n &= \frac{1}{\log Q} \left((Q+1 - \frac{1}{Q}) \log n \right. \\ &\quad \left. + (Q+1 - \frac{1}{Q} H_{b-1} + \frac{Q-1}{Q^2} b) \right. \\ &\quad \left. + P(\log n) + O(\frac{1}{n}) \right), \end{aligned}$$

where $H_k = \sum_{i=1}^k \frac{1}{i}$ denotes the harmonic sum, and $P(x)$ is a periodic function of period $\log Q$ with very small amplitude.

Proof – When $n \leq b$, then $D_n = n$ and $S_n = n^2$, since the database is exchanged as a whole in this case.

When $n > b$, elementary algebra on random partitions yields:

$$S_n = nQ + \sum_{n_1 + \dots + n_Q = n} Q^{-n} \binom{n}{n_1 \dots n_Q} (S_{n_1} + \dots + S_{n_Q}).$$

Denoting $S(z) = \sum_n S_n \frac{z^n}{n!} e^{-z}$ the so-called Poisson generating function of S_n , we get the functional equation:

$$\begin{aligned} S(z) &= QS\left(\frac{z}{Q}\right) + Qz - \left((Q+1 - \frac{1}{Q}) z e_b(z) \right. \\ &\quad \left. + \frac{(Q-1)}{Q^2} z^2 e_{b-1}(z) \right) e^{-z} \end{aligned}$$

with the convention that $e_k(z) = \sum_{i \leq k} \frac{z^i}{i!}$. Let $D(z) = S(z)/z$. We therefore have the functional equation:

$$\begin{aligned} D(z) &= D\left(\frac{z}{Q}\right) + Q - \left((Q+1 - \frac{1}{Q}) e_b(z) \right. \\ &\quad \left. + \frac{(Q-1)}{Q^2} z e_{b-1}(z) \right) e^{-z} \end{aligned}$$

Using the Mellin transformation $D^*(s) = \int_0^\infty D(x) x^{s-1} dx$, defined for $\Re(s) \in]-1, 0[$, and using the fact that the Mellin transformation of $D(\frac{z}{Q})$ is $Q^s D^*(s)$, we get to the closed form solution:

$$D^*(s) = \frac{\Gamma_b(s)(Q+1 - 1/Q) + \Gamma_{b-1}(s+1)(Q-1)/Q}{1 - Q^s}$$

with the convention that $\Gamma_k(s)$ is the Mellin transformation of $e_b(z)e^{-z}$. We

note by the way that, $\Gamma_k(s) = \sum_{i \leq k} \frac{\Gamma(s+i)}{i!}$.

The reverse Mellin transformation then yields:

$D(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} D^*(s) \exp(s \log x) ds$ for any c such that $\Re(c) \in]-1, 0[$. When the integration path moves to the right, it encounters a succession of singularities on the vertical axis $\Re(s) = 0$. There is a double pole on $s = 0$ and there are single poles on $s_k = \frac{2ik\pi}{\log Q}$ for k being integer. Therefore, by virtue of singularity analysis, we have for any m :

$$\begin{aligned} D(x) = & -\mu_0 \log x - \lambda_0 \\ & - \sum_k \lambda_k \exp(-2ik\pi \frac{\log x}{\log Q}) \\ & + O(\frac{1}{x^m}), \end{aligned}$$

where λ_0 and μ_0 are, respectively, the first and second order residus of $D^*(s)$ at $s = 0$, where and λ_k is the first order residus at $s = s_k$. Identifying residus is a trivial matter. Notice that $P(x) = -\sum_k \lambda_k \exp(-2ik\pi \frac{\log x}{\log Q})$, which is periodic of period $\log Q$. The estimate is true for every m , since there are no more singularities in the right half plan.

We use the depoissonization theorem to assess that $S_n = nD_n = nD(n) + O(1)$ when $n \rightarrow \infty$, which terminates the proof of the theorem.

Figure 1.1 shows the asymptotic behaviour of retrieval cost with $Q = b = 16$ for n varying from 100 to 1,000. It is compared with full database retrieval cost.

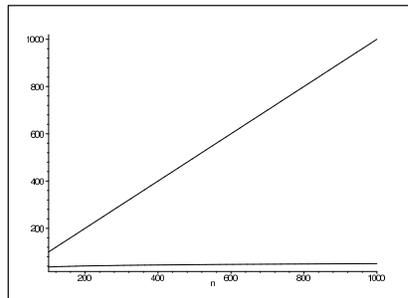


Figure 1.1: Signature retrieval cost (bottom) compared with full database retrieval cost (top) with a single record mismatch versus database size: $Q = 16$, $b = 16$

1.5 Applicability of the Database Signature Exchange Mechanism

This section outlines the applicability of the specified mechanisms in a set of common scenarios. One application has been discussed previously: ensuring that information from LSA messages, originating from attached wired networks with potentially long intervals between LSA message generation, is maintained in all nodes in the wireless ad-hoc network. The scenarios outlined in this section go beyond that situation, and consider how the database signature exchange may apply even in pure wireless scenarios.

1.5.1 Emerging Node

When a new node emerges in an existing network, the initialisation 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 initialisation time, the database exchange mechanisms can be employed as soon as the node has established a relationship with one neighbour node already initialised. This emerging node will select a neighbour 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 initialisation time of new nodes in the network.

1.5.2 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 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.

1.5.3 Reliable Flooding

If a node wants a specific LSA to be reliably transmitted to its neighbour, 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 neighbour 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.

1.6 Conclusion

In this paper, we have introduced the notion of database exchange and reliable synchronisation in the context of wireless ad-hoc networks. Inspired by the mechanisms from the routing protocol OSPF, we have argued that in the form, present in OSPF, these mechanisms are not suitable for the wireless ad-hoc domain. While OSPF is designed for relatively static networks, the potentially very dynamic nature of wireless ad-hoc networks imply that database inconsistencies may arise more frequently with less available network capacity for alleviating the inconsistencies – and that acknowledgement-based reliability is unsuitable since the correct interpretation of an acknowledgement depends on being received in a specific context.

Consequently, we have devised a mechanism for database exchange, adapted for the the specific environment of wireless ad-hoc networks. The mechanism is proposed as an extension to the OSPF interface type WOSPF [1]. The mechanism allows an efficient way of detecting and alleviating database inconsistencies, and can furthermore be employed as a way of providing “context-independent selective acknowledgements” for reliable synchronisation and link-state diffusion.

We have, analytically, compared the performance of our proposed mechanism to the performance of the mechanisms for database exchange in OSPF, and found it to be superior in terms overhead. We have furthermore outlined a couple of scenarios, where application of the mechanisms devised in this paper may be advantageous for a wireless ad-hoc network.

Ongoing and future work on this topic involves extending the analytical performance evaluation of this mechanism, as well as conducting exhaustive simulations and experimental testing, comparing the performance of WOSPF with or without database exchange and reliable synchronisation mechanisms.

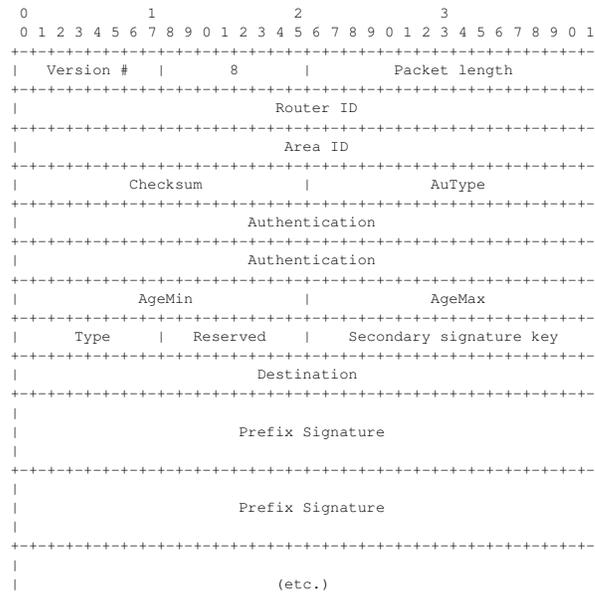
Bibliography

- [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.
- [8] A. Qayyum, L. Viennot and A. Laouiti “Multipoint relaying: An efficient technique for flooding in mobile wireless networks”, INRIA research report RR-3898.

Packet Formats

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

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 1.3.3.

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 1.3.1.

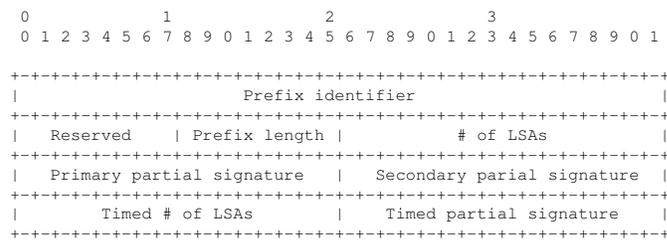
Destination

- 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 zeroed.

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 1.3.3.

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, 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 hashing 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 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.