OSPF-style Database Exchange and Reliable Synchronization in the Optimized Link-State Routing Protocol

T. Clausen, E. Baccelli, P. Jacquet
T.Clausen@computer.org, {Emmanuel.Baccelli, Philippe.Jacquet}@inria.fr
Project Hipercom, INRIA Rocquencourt
BP 105, 78153 Le Chesnay Cedex, France
Phone: +33 1 3963 5133 Fax: +33 1 3963 5566

Abstract— **The Optimized Link-State Routing protocol (OLSR) is a proactive link-state routing protocol. While similar to the well-known Internet routing protocol OSPF, OLSR is designed to be simple, and to maintain connectivity in face of highly dense and dynamic networks, while being ressource-economic (battery, bandwidth etc.) These characteristics make OLSR suitable as an underlaying routing protocol in a wide range of ad-hoc sensor networks.**

In this paper, we introduce an extension to OLSR: OSPF-style database exchange and reliable synchronization. The goal of this extension is to provide a mechanism, through which nodes in an ad-hoc sensor network can detect and correct discrepancies in their link-state databases. We qualify why the mechanism, found in OSPF, is not directly applicable for ad-hoc sensor networks, describe an adopted mechanism, accomplishing the same goal, and evaluate the performance of this mechanism in comparison to the database exchange mechanism found in OSPF. We finally discuss some applications of database exchange and reliable synchronization in ad-hoc sensor networks.

I. INTRODUCTION

The Optimized Link-State Routing Protocol (OLSR) [3] is a proactive link-state routing protocol. OLSR shares many algorithmic similarities to OSPF [1], the predominant IGP deployed in the Internet. OLSR is designed to maintain network connectivity in face of frequent node movements and high network dynamics, and to operate under the constraint of ad-hoc sensor networks (processing power, bandwidth, battery capacity etc.)

Inspired by the similarities between OLSR and OSPF, this paper discusses an extension to OLSR, providing OSPF-style database exchange and reliable synchronization. These mechanisms serve the goal of ensuring that discrepancies between the topology databases in the nodes in the network are detected and, eventually, elliminated.

The database exchange and reliable synchronization mechanisms, in the form found in OSPF, are not suitable for ad-hoc and sensor networks: performing a database synchronization requires transmission of complete link-state databases, even in the presence of only a few discrepancies. Furthermore, the potential topology dynamics of ad-hoc sensor networks implies that some links in the network may change frequently and, therefore, that by the time a database synchronization is completed, a new discrepancy has emerged. I.e., in a dynamic network, database synchronization may become an ongoing, expensive, process.

In this paper we propose a mechanism, adapted for the low-bandwidth high-dynamics conditions of ad-hoc sensor networks. We specify the mechanism in terms of the OLSR protocol, but notice that it may be applicable for any proactive link-state routing protocol, including OSPF. 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 ad-hoc sensor network, reduced initialization time when new nodes are emerging in the network and reduced overhead and reduced convergence time when several network clouds merge.

The remainder of this paper is therefore organized as follows: section II contains a brief description of OLSR, followed by a discussion of the database exchange mechanisms of OSPF in section III. Following, section IV develops a mechanism, for database exchange and reliable synchronization in OLSR, specifically designed for the characteristics of ad-hoc sensor networks. Section V evaluates the designed mechanism, and compares with the performance of the database exchange mechanism from OSPF. Section VI discusses different applications of the developed mechanism and section VII concludes the paper.

II. OLSR Overview

The Optimized Link State Routing protocol (OLSR) [3], is a proactive link state routing protocol, designed specifically for ad-hoc networks. OLSR consists of three components: (i) a neighbor sensing mechanism, through which a node manages its local topology; (ii) an optimized flooding mechanism, dramatically reducing the overhead of flooding operations and (iii) a link state diffusion mechanism, through which nodes manage global topology information. These components are described briefly in the following.

A. Neighbor Sensing

OLSR performs neighbor sensing through periodic exchange of HELLO messages between neighbor nodes. A HELLO message encodes the emitting nodes own address, as well as a list of (addresses, status) pairs for neighbor nodes, known by the originator of the HELLO message. This allows a node to verify bidirectionality of links to neighbors, as well as supplies each node with a list of its symmetric neighbors, symmetric 2-hop neighbors and MPR selectors. This information is used by the MPR selection, described in section II-B.

B. Multipoint Relay Selection and Signaling

An optimized flooding mechanism is employed by OLSR, defined by the concept Multipoint Relays (MPRs): each node must select MPRs from among its neighbor nodes such that a message emitted by a node and repeated by the MPR nodes will be received by all nodes, two hops away. The neighborhood and 2-hop neigborhood information is provided by the neighbor sensing mechanism. The neighbor sensing mechanism is also used by a node to signals its MPR selection – which in turn allows each node to maintain an *MPR selector set*, describing which nodes have selected it as MPR.

MPR flooding, then, works as follows: a flooded message is relayed by a node iff: (i) the last-hop of the control message is an MPR selector and (ii) if the message has not been previously received by the node. Figure 1 shows a node with neighbors and 2-hop neighbors, the arrows indicating the transmissions required to flood a message from the center node.

For further information, including an efficient heuristic for computing the MPR set of a node, refer to [4].

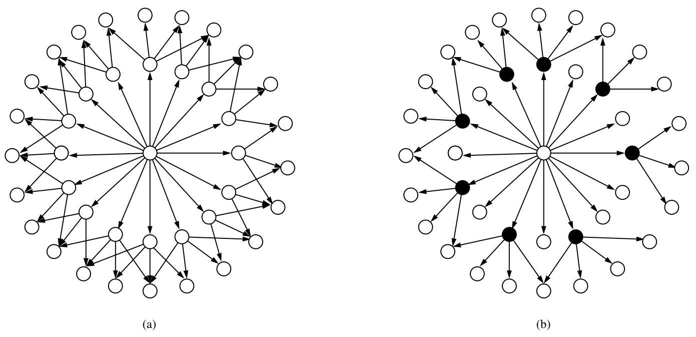

Fig. 1. Two hop neighbors and "multipoint relays" (the solid circles) of a node. (a) illustrates the situation where all neighbors retransmit a broadcast, (b) illustrates where only the MPRs of a node retransmit the broadcast.

C. Link State Diffusion

The objective of link-state diffusion is, that each node in the network receives link-state information pertaining to each other node in the network. In OLSR, this is accomplished through TC-messages. A TC-message contains a set of bi-directional links between a node and (a subset of) its neighbors[1]. TC messages are diffused to the entire network, employing MPR flooding mechanism, see section II-B. Furthermore, only nodes, selected as MPR, generate TC-messages.

III. Database Exchang and Reliable Synchronization in OSPF

OSPF [1] employs two independent mechanisms for maintaining globally consistent topology information: (i) reliable transport of LSA messages and (ii) database exchanges between pairs of routers. Functionally, LSA messages in OSPF are identical to TC messages in OLSR, the key difference being, that OLSR employs unreliable, periodic flooding of TC messages and OSPF employs reliable transport of LSA messages.

A. Reliable Transmission

OSPF employs positive acknowledgments (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

[1]For a discussion on the selection of which neighbors to include in the TC messages in order to provide sufficient topology information, refer to [3] and [5].

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.

1) Reliable Transmission and ad-hoc sensor networks: In ad-hoc sensor 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 neighbors 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 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. 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 an ad-hoc sensor network, where the network may be substantially more dynamic: nodes may move out of range etc.

B. Database Exchange

OSPF database exchanges are intended to synchronize 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.

1) Database Exchange and ad-hoc sensor networks: In the context of ad-hoc sensor 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.

IV. Database Signature Exchange

In this section, we propose a mechanism for database exchange and reliable synchronization, adapted to ad-hoc sensor networks. Specifically, we propose an extension to OLSR.

The basic idea 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.

Our approach is somewhat inspired by IS-IS [2], in which packets which list the most recent sequence number of one or more LSAs (Sequence Numbers packets) are used to ensure that neighboring 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 acknowledgment 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 TCs issued from within a very dynamic part of an ad-hoc sensor network (e.g. RFID tags on products in a plant): TCs from nodes within this domain should transmitted frequently and periodically, thus information describing these nodes is frequently updated and “with a small age”. TC messages from a less mobile part of the ad-hoc sensor network (e.g. sensors on semi-permanent installations in the plant) might be updated less frequently. Thus consistancy 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.

A. Definition of Link State Database Signatures

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

$$\begin{aligned} Signature\ Message &= (Age\ Interval, Key, \\ &\quad Prefix\ Signature), \end{aligned}$$

A signature features a set of prefix signatures:

$$Prefix\ Signature = (Prefix, Sign(Prefix)).$$

Each $Sign(Prefix)$ 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 $Sign(Prefix)$ has the following structure:

$$\begin{aligned} Sign(Prefix) &= (Primary\ Partial\ Signature, \\ &\quad Secondary\ Partial\ Signature, \\ &\quad Timed\ Partial\ Signature, \\ &\quad \#TC, Timed\ \#TC). \end{aligned}$$

A primary partial signature (PPS) for a prefix is computed as a sum over all TCs in a nodes link-state database, where the prefix matches the advertising router of the TC:

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

$\sum_{prefixes}$ denotes the sum over prefixes matching the advertising router of the TC. The secondary partial signature (SPS) for a prefix is computed as a sum over all TCs in a nodes link-state database, where the prefix matches the advertising router of the TC:

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

$\sum_{prefixes}$ denotes the sum over prefixes matching the advertising router of the TC. The timed partial signature (or TPS) for a prefix and an age interval is computed over TCs in a nodes link-state database where:

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

and has the following expression:

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

$\sum_{prefixes,time}$ denotes the sum over prefixes matching the advertising router of the TC and where the age falls within the age interval of the advertisement. The TC identifier is the string, obtained through concatenating the following fields from the OLSR message, encapsulating the TC message, as well as the TC message header [3]: Originator Address, ANSN (Advertised Neighbor Sequence Number).

B. Signature Exchange

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

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 IV-C details how info signatures are generated; section IV-D details how signatures are employed to detect link-state database discrepancies.

2) Database Signature Exchange: Database exchange (dbx) signatures are directed towards a single neighbor only. The purpose of emitting a dbx signature is 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 of its neighbors, 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 neighbors. The node builds a dbx message signature, based on the information acquired from the info signature exchange.

C. Signature message generation

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

1) Info Signatures: An info signature message describes the complete link state database of the node that sends it. Abcense of information in a signature indicates abcense 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 TCs 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:

$$\begin{aligned} Weight(prefix) &= \#\ of\ LSAs\ whose \\ &\quad originator\ matches \\ &\quad the\ prefix. \end{aligned}$$

And similarly, the timed weight as the function:

$$Timed\,Weight(prefix) \;=\; \#\;of\;TCs\;whose\;originator\;matches\;the\;prefix\;and\;whose\;age\;falls\;inside\;the\;age\;interval.$$

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) Dbx Signatures: Dbx signatures serve to trigger an exchange of descrepant TCs with one neighbor, known to have more up-to-date link-state information – the ideal is to pick the neighbor 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. In OLSR, 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.

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 use the recursive algorithm described in section IV-C.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 TCs.

D. Checking Signatures

Upon receiving a signature message from a neighbor, a node can check its local TC database[2] 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:

1) both the number of TCs and the timed number of TCs 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, 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.

E. 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 TC 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 TC 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 OLSR packet, containing the missing TC's. These TC's must be transmitted with their TTL equal to 1 (one hop only). These TCs must

[2]In the OLSR specification, the TC database is described in form of a "Topology Set", recording the information received through TC message exchange.

indicate the age featured at the moment in the database, from which they are taken.

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

V. 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.

We denote by n the number of records in the database (typically $n \approx 1,000$, and by Q the number of aggreged 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.

A. 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

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

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.

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:

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

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:

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

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:

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

Using the Mellin transform $D^*(s) = \int_0^\infty D(x)x^{s-1}dx$, defined for $\Re(s) \in]-1,0[$, and using the fact that the Mellin transform of $D(\frac{z}{Q})$ is $Q^s D^*(s)$, we arrive 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 transform of $e_b(z)e^{-z}$. In passing, $\Gamma_k(s) = \sum_{i \leq k} \frac{\Gamma(s+i}{i!}$

Reverse Mellin then yields:

$D(x) = \frac{1}{2\mathrm{i}\pi} \int_{c-\mathrm{i}\infty}^{c+\mathrm{i}\infty} D^*(s) \exp(s \log x) ds$ for any c such that $\Re(c) \in]-1,0[$.

When the integration path move to the right, it encounters a succession of singularities on the vertical axis $\Re(s) = 0$. There is a double pole on on $s = 0$ and there are single poles on $s_k = \frac{2\mathrm{i}k\pi}{\log Q}$ for k being integer. Therefore, by virtue of singularity analysis, we have for any m:

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

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 $P(x) = -\sum_k \lambda_k \exp(-2\mathbf{i}k\pi \frac{\log x}{\log Q})$, 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 asses that $S_n = nD_n = nD(n) + O(1)$ when $n \to \infty$, which terminates the proof of the theorem.

Figure 2 shows the asymptotic behavior of retrieval cost with $Q = b = 16$ for n varying from 100 to 1,000. It is compared with full database retrieval cost.

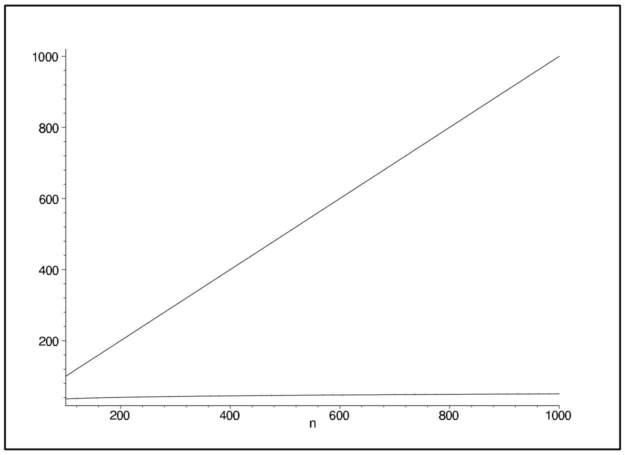

Fig. 2. signature retrieval cost (bottom) compared with full database retrieval cost (top) with a single record mismatch versus database size: $Q = 16$, $b = 16$

VI. 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, originating from attached wired networks, which potentially is injected to the ad-hoc sensor network only infrequently, is maintained in all nodes. The scenarios outlined in this section go beyond that situation.

A. 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 TCs are not transmitted frequently (outside TCs would be an example of such), this mechanism may drastically reduce the initialization time of new nodes in the network.

B. Merging Wireless Clouds

Two disjoint sets of nodes, employing OLSR 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 TC-D mechanism, the convergence time until a new stable state is achieved can be kept at a minimum.

C. Reliable Flooding

If a node wants a specific TC messate 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 TC message broadcasts an info signature, containing the full TC originator address as signed prefix and a very narrow age interval, centered on the age of the TC which is to be reliably transmitted. A neighbor which does not have the TC in its database will therefore automatically trigger a database exchange concerning this TC, and send a dbx signature containing the TC-originator address signed with an empty signature. The receiving of such a dbx signature will trigger the first node to retransmit the TC right away, to ensure that the TC message does get through.

VII. Conclusion

In this paper, we have introduced the notion of database exchange and reliable synchronization in the context of ad-hoc sensor 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 ad-hoc sensor domain. While OSPF is designed for relatively static networks, the potentially very dynamic nature of ad-hoc sensor networks imply that database inconsistencies may arrise 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.

Concequently, we have deviced an mechanism for database exchange, adapted for the the specific environment of ad-hoc sensor networks. The mechanism is proposed as an extension to the routing protocol OLSR. The mechanism allows an efficient way of detecting and alleviating database inconsistencies, and can furthermore be employed as a way of providing "context-independant selective acknowledgements" for reliable synchronization 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 superiour in terms overhead. We have furthermore outlined a couple of scenarios, where application of the mechanisms deviced in this paper may be advantageous for an ad-hoc sensor 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 OLSR with or without database exchange and reliable synchronization mechanisms.

References

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

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

[3] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol," RFC 3626, http://ietf.org/rfc/rfc3626.txt, 2003.

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

[5] T. Clausen, P. Jacquet, L. Viennot, "Investigating the Impact of Parital Topology in Proactive MANET Routing Protocols", Fifth International Symposium on Wireless Personal Multimedia Communications 2002 Proceedings.