Epidemic Dissemination in Ad Hoc Networks

Lidia Yamamoto

Hitachi Europe, Sophia Antipolis Laboratory
1503 Route des Dolines, F-06560 Valbonne, France
Lidia.Yamamoto@hitachi-eu.com

Abstract. We have worked on an epidemic dissemination protocol to maintain soft-state in a decentralized, peer-to-peer fashion, in ad hoc networks. This protocol is an enhancement of Passive Distributed Indexing (PDI) method proposed by Lindemann and Waldhorst. We have enhanced PDI in order to reduce the number of broadcast messages when the search for an item may span several hops. Three enhancements are proposed: *(i)* Lazy query propagation to delay the propagation of query messages such that local responses can inhibit unnecessary search. *(ii)* Quench waves to stop an already initiated query propagation when still possible. Decision rules based solely on local information determine whether to start a quench wave or not. *(iii)* The use of Multi-Point Relay (MPR) or similar protocol and algorithm, to reduce redundant broadcast messages.

Keywords: ad hoc network, epidemic dissemination, peer-to-peer

1 Introduction

Peer-to-Peer (P2P) and ad hoc networks have many points in common: both represent a decentralized self-organizing network structure. However few existing P2P algorithms are specifically designed to operate efficiently over ad hoc networks. And few ad hoc networks are designed to benefit from P2P infrastructures.

We have worked on an epidemic dissemination protocol to maintain soft-state in a decentralized, peer-to-peer fashion, in ad hoc networks. This protocol is an enhancement of Passive Distributed Indexing (PDI) method proposed by Lindemann and Waldhorst [1, 2]. PDI is a method for distributing information in a P2P structure which is particularly suited to ad hoc networks, and does not involve an overlay topology. It makes use of broadcast messages to spread information via passive epidemic dissemination.

We have enhanced PDI in order to reduce the number of broadcast messages when the search for an item may span several hops. Three enhancements are proposed:

- Lazy query propagation to delay the propagation of query messages such that local responses can inhibit unnecessary search.

Dagstuhl Seminar Proceedings 04411
Service Management and Self-Organization in IP-based Networks
http://drops.dagstuhl.de/opus/volltexte/2005/101

- Quench waves to stop an already initiated query propagation when still possible. Decision rules based solely on local information determine whether to start a quench wave or not.
- The use of Multi-Point Relay (MPR) or similar protocol and algorithm, to reduce redundant broadcast messages.

In this memo we present the current state of this research, open aspects and future directions.

2 Background

2.1 Passive Distributed Indexing (PDI)

Passive Distributed Indexing (PDI) [1,2] is a general-purpose distributed lookup method for $< key, value >$ pairs. The PDI concept was introduced in [1], which also describes the mechanisms for multi-hop query propagation, response, and caching of query results. PDI is enhanced in [2] with epidemic dissemination and invalidation techniques for keeping the cache entries coherent.

PDI is designed for mobile networks. It can benefit from node mobility and underlying broadcast mechanisms available for mobile networks for the dissemination of information to peer-to-peer nodes. Each node holds an index cache which is a list of $< key, value >$ pairs. A query is a message from a node that wishes to obtain the value for a given key. Queries are sent in broadcast to all one-hop neighbours of the querying nodes. Responses are also sent in broadcast to all one-hop neighbours of the responding node. This implies that even if a node is not looking for a particular answer to a query, it listens to on-going responses and updates its internal index cache with the observed $< key, value >$ pair. Moreover, nodes moving to other regions in the network carry their values in their local cache and broadcast query responses in the new zone. In this way, popular entries will be disseminated to several nodes. This mechanism is called “epidemic dissemination” because it works like an infectious disease: “infected” nodes “contaminate” other nodes with information they have in their local caches. PDI also supports query propagation over multiple hops. The maximum number of hops, or query TTL (time-to-live), is specified by the inquiring node. Responses may also be forwarded over multiple hops.

ORION (Optimized Routing Independent Overlay Network) [3] is another approach to peer-to-peer networking in which a response message navigates through the reverse path taken by the corresponding query message. ORION also provides a method to filter responses along the reverse path, in order to remove redundant responses.

2.2 Multi-Point Relaying (MPR)

Multi-Point Relaying (MPR) [4,5] is a technique to decrease the number of broadcast message copies generated in a mobile network, while at the same time ensuring that all nodes in the network will receive at least one copy of the broadcast message. It consists in identifying redundant nodes in the broadcast distribution graph and preventing those nodes from relaying broadcast messages, thereby reducing the number of redundant transmissions. The authors have proved that the computation of the minimum MPR set is an NP-complete problem and propose a simple and efficient heuristic to compute an MPR set that in practice is close enough to the minimum.

3 Additional enhancements to PDI

PDI has advantages and shortcomings. The advantages are that it is simple, requires no routing protocols, requires no network overlays, is robust to network partitioning and failures, and exploits locality: information spreads where it is more popular, i.e. where there is more demand for it. The shortcomings are the redundant propagation of queries and replies, the dependence upon node movement for epidemic dissemination, and the reliance upon query locality or large query TTL for successful information retrieval.

We would like to improve PDI in order to reduce the number of redundant messages, and to decrease the dependence upon node movement and query locality. As initial steps, we have devised three improvements to PDI aimed at decreasing the number of broadcast messages:

- The use of existing MPR protocol and algorithm: when a node M receives a query from neighbour node N, node M is only allowed to propagate the query to other nodes if node M is node N's MPR.
- Lazy query propagation: delay/listen before broadcast relaying – a node that receives a query starts a timer and listens to the responses of others: if the node overhears a response for the queried item within the timer period, then the node does not propagate the query further.
- Quench wave generation: after querying node receives response, a decision algorithm determines whether it should start a quench wave. The quench wave propagates as quickly as possible and seeks to stop on-going query broadcasts.

We now describe each of these enhancements in more detail.

3.1 Use of MPR within PDI

The most obvious enhancement to PDI is to use an algorithm to minimize the number of redundant broadcasts. One of the most popular algorithms in this category is MPR, briefly described in Section 2.2.

There are several ways to exchange messages in order to obtain the neighbour information necessary to calculate the MPR set. Within the OLSR protocol [6, 7] the MPR set is built when exchanging HELLO messages, which contain neighbour information also used for routing purposes. This protocol could be used, but other protocols are also possible, for example a dedicated one to determine MPR information without being coupled to OLSR. Moreover, the MPR algorithm and protocol could also be integrated within the Enhanced PDI implementation. However this could introduce redundant messages when OLSR or other protocols or services that make use of MPR are also running in the same network. Here we fall into a typical service composition problem.

3.2 Lazy Query Propagation

In the original PDI method, queries are propagated over the number of hops determined by the inquiring node in the form of a ttl_{query} field in the query message. ttl_{query} determines the maximum number of hops a given query may traverse. After traversing ttl_{query} hops the query message is discarded. In order to reduce unnecessary query propagation, query messages are propagated in a "lazy" way. This is called the Lazy Query Propagation mechanism. It works as follows:

A node that receives a query (and is allowed to propagate it according to the MPR criterion) and has no answer for it in its local PDI entry cache, waits for a period dw_{query} in which it listens for possible responses to the query. If within the period dw_{query} the node overhears a response to the query, it caches the response (epidemic dissemination) and takes no further action concerning the query. If, on the other hand, the node does not hear any response to the query within dw_{query}, after the interval dw_{query} has passed, the node re-broadcasts the query to its immediate neighbours. In this way, only the queries for which no response is available in the vicinity are propagated to other regions of the network.

3.3 Quench Wave

After an inquiring node receives a reply to its query, the query and response messages that might still be in transit are useless. A lot of useless traffic can thus be avoided if we are able to stop the query and response propagation flows in time. If the node receives a reply sufficiently early, it is still possible to attempt to stop these messages from propagating further. If the inquiring node decides that it is still time to stop the messages, it broadcasts a Quench Message. In contrast to the Query Message, the Quench Message is not delayed at the relay nodes. It is propagated as quick as possible in order to stop as many Query and Response Propagation as possible. We call the resulting multi-hop broadcast of Quench Messages a Quench Wave. Upon reception of a Quench Message, a node cancels any corresponding delayed query propagation.

Issuing Quench Messages at a too late stage can contribute to flood the network and cause further congestion, while at the same time being ineffective in stopping already propagated messages. The usefulness of a Quench Wave is therefore limited to a short period of time.

We now show how the inquiring node decides whether to issue a Quench wave or not, after receiving a response to its query. Let d_r be the elapsed time from the moment the inquiring node issues a query message to the moment it receives a corresponding response message. Let d_t be the average time a message takes to go from one hop to the other. Given an arbitrary node N_0 and one of its one-hop neighbours N_1, d_t from N_0 to N_1 includes the queuing delay at the outgoing interface of N_0 towards N_1, plus the link propagation delay from N_0 to N_1, plus any processing delay at N_1. Recall from Section 3.2 that dw_{query} (from now on we call it d_w for short) is the delay that a relay node waits before considering the possibility of further propagating a query message.

The quench wave is able to stop the query wave when the former meets the outermost border of the latter. The query wave is slower to propagate than the quench wave, due to the waiting time d_w. On the other hand the quench wave cannot start earlier than d_t after the query wave. Using d_r, d_t, and d_w it is possible to obtain the value h, which is the distance in hops from the inquiring node to the zone where the two waves meet. This can be calculated as follows:

$$t = d_r + d_t \cdot h = (d_w + d_t) \cdot h \Rightarrow h = \frac{d_r}{d_w} \tag{1}$$

Where t is the time necessary for the two waves to meet.

Therefore it suffices to know d_r and d_w in order to calculate h (d_t is not necessary). The value of d_r can be easily measured at the inquiring node by reading the current time at the moment the query is sent, and then reading it again when the response is received: d_r is simply the interval between query and response. As for d_w, it can be a parameter of the protocol, with a well-known value. Relay nodes could use d_w as the waiting delay, or a random delay with average d_w (in order to avoid synchronization of query propagation messages and consequent risk of congestion).

The decision consists in setting a threshold beyond which it is considered harmful to send quench messages. Let h_{max} be this threshold. Then the decision algorithm is expressed by a very simple rule, as follows:

If $h < h_{max}$ then issue quench else do nothing.

The question now is how to estimate the optimum h_{max} in order to ensure that there is a benefit from sending a quench wave when compared to not sending it. The benefit exists when the quench wave results in reduction of the total number of broadcast messages corresponding to the query (query plus quench messages).

An estimation of the potential number of query and quench messages depends on the query TTL, number of nodes, position and distribution of nodes, and so on. Assuming uniform node density on a bi-dimensional area, we first calculate the number of messages for the ideal case where no packet loss occurs.

When no quench wave is generated, query messages propagate until their TTL expires. Therefore, the total number of messages when no quench wave is generated is proportional to the area of a circle delimited by the maximum query TTL q_{max}:

$$m_n = 1 \cdot c \cdot q_{max}^2 \tag{2}$$

where c is a constant that depends on the node density.

When a quench wave is generated, the total number of messages corresponds to the number of query messages plus the number of quench messages. Since both waves meet at h_{max}, and assuming successful quench (i.e. no "leaks" as it will be explained later in this document) the number of quench messages is roughly the same as the number of query messages, and both are proportional to the area of the circle of radius h_{max}. Therefore, the total number of messages when a quench is generated can be expressed as:

$$m_q = 2 \cdot c \cdot h_{max}^2 \tag{3}$$

The ratio between m_q and m_n can then be easily calculated, as follows:

$$\frac{m_q}{m_n} = \frac{2 \cdot h_{max}^2}{q_{max}^2} \tag{4}$$

An improvement occurs when the quench wave results in less messages than the case without quench:

$$m_q < m_n \Rightarrow \frac{m_q}{m_n} < 1 \Rightarrow \tag{5}$$

$$\frac{2 \cdot h_{max}^2}{q_{max}^2} < 1 \Rightarrow \tag{6}$$

$$h_{max} < \frac{1}{\sqrt{2}} \cdot q_{max} \tag{7}$$

Therefore in the ideal situation (no packet loss) we can set h_{max} to approximately 70% of the maximum query TTL q_{max}. The latter is assumed as given, as a parameter of the protocol.

Packet losses are common in wireless networks, therefore the ideal situation is not likely to occur under realistic conditions. We now study the impact of packet losses on the usefulness of the quench mechanism. We define a leak as a query message that is not successfully quenched, and therefore continues to propagate until its TTL expires. In the worst case, the leak will propagate $q_{max} - h_{max}$ hops until it is discarded.

We have calculated the ratio of total number of messages with quench over the total without quench (mq/mn), for k equidistant leaking points, and still assuming uniform distribution of nodes over a 2-D surface. The calculation (not shown) uses simple geometric concepts to estimate the areas flooded from the leaking points until the query TTL expires. Figure 1 shows the result for up to five equidistant leaking points. The quench wave leads to improvement (decrease in the number of broadcast messages) when $mq/mn < 1$). For up to 3 leaking points there is still some gain from the quench mechanism. For more than 3 leaking points the curves remain mostly on the upper side ($mq/mn > 1$) therefore there is no benefit from the quench mechanism. This result helps to refine the decision mechanism of when to issue the quench wave, based on some estimation on the loss probability of the wireless links.

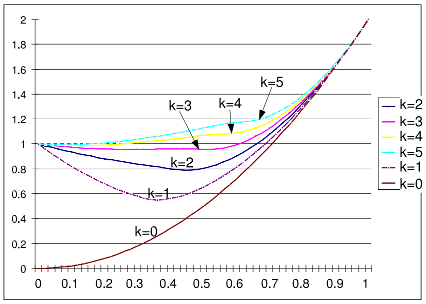

Fig. 1. Ratio of messages without quench over that with quench (mq/mn on the y axis), for k equidistant leaking points. The x axis is the ratio h_{max}/q_{max}.

4 Current state and future work

We are currently refining the epidemic dissemination mechanism, and implementing it both on a simulated environment and in a wireless ad hoc network testbed. More analysis is needed to fully understand the locality effects of such epidemic protocol with soft-state parameters. We are investigating ways to apply epidemic modelling from biology for this analysis. We would like to study combined techniques to improve protocol efficiency, such as hybrid gossip and passive dissemination methods, or hybrid use of distributed hash tables (DHT) with epidemic dissemination. Another issue that deserves more attention is the aggregation of queries and responses in order to reduce network load, as in ORION. It is also important to enhance the protocol with security features to prevent denial of service and other attacks based on fake response and quench messages.

Acknowledgments

I would like to express my gratitude towards my colleagues from Hitachi and Hitachi Europe who made this work possible, especially Mr. Masato Hayashi and Mr. Tatsuaki Osafune who provided highly valuable guidance and help.

References

1. Lindemann, C., Waldhorst, O.: A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. In: Proceedings of the 2nd IEEE Conference on Peer-to-Peer Computing (P2P 2002), Linköping, Sweden (2002) 71–83
2. Lindemann, C., Waldhorst, O.: Consistency Mechanisms for a Distributed Lookup Service supporting Mobile Applications. In: Proceedings of the 3rd Int. ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE 2003), San Diego, CA (2003)

3. Klemm, A., Lindemann, C., Waldhorst, O.: A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks. In: Proceedings of IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL, USA (2003)
4. A.Qayyum, Viennot, L., Laouiti, A.: Multipoint Relaying: An Efficient Technique for flooding in Mobile Wireless Networks. Technical Report Research Report RR-3898, INRIA, France (2000)
5. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. In: Proceedings of 35th Annual Hawaii International Conference on System Sciences (HICSS'2002). (2002)
6. Clausen, T., Jacquet, P., Laouiti, A., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized Link State Routing Protocol. In: IEEE INMIC, Pakistan (2001)
7. Clausen, T., Jacquet (Editors), P.: Optimized Link State Routing Protocol (OLSR). Internet rfc 3626 (category: Experimental), IETF (2003)