Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in Large-Scale Sensor Networks

Xin Liu
Department of Computer Science
University of California
Davis, CA 95616, USA
Emails: liu@cs.ucdavis.edu

Qingfeng Huang and Ying Zhang
Palo Alto Research Center (PARC) Inc.
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Emails: {qhuang, yzhang}@parc.com

ABSTRACT

In this paper we investigate efficient strategies for supporting on-demand information dissemination and gathering in large-scale wireless sensor networks. In particular, we propose a "comb-needle" discovery support model resembling an ancient method: use a comb to help find a needle in sands or a haystack. The model combines push and pull for information dissemination and gathering. The push component features data duplication in a linear neighborhood of each node. The pull component features a dynamic formation of an on-demand routing structure resembling a comb. The comb-needle model enables us to investigate the cost of a spectrum of push and pull combinations for supporting discovery and query in large scale sensor networks. Our result shows that the optimal routing structure depends on the frequency of query occurrence and the spatial-temporal frequency of related events in the network. The benefit of balancing push and pull for discovery in large scale geometric networks are demonstrated. We also raise the issue of query coverage in unreliable networks and investigate how redundancy can improve the coverage via both theoretical analysis and simulation. Last, we study adaptive strategies for the case where the frequencies of query and events are unknown *a priori* and time-varying.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Protocols, Wireless Communications.

General Terms

Algorithms, Design

Keywords

Query, routing, data dissemination, sensor network, pull, push

SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.

1. INTRODUCTION

The goal of this work is to investigate efficient and reliable routing mechanisms for supporting queries in large scale ad hoc wireless sensor networks. Recently, wireless sensor network has attracted a lot of attention from the research community, e.g., [1, 2, 3, 4, 5, 6, 7, 8]. Many emerging sensor network applications involve dissemination of observed information to interested clients and thus demand efficient dissemination mechanisms due to resource limitations. For instance, a sensor network might be deployed for enhancing soldiers' battle field situation awareness when visibility is low (either at night or due to smoke). A solider might be interested in where the tanks are in the battlefield. The nodes detecting the tanks can periodically push (broadcast) the information throughout the network in anticipation of the soldier's needs, as shown in Figure 1(a). This push-based information dissemination strategy is efficient when there are many solders in the network constantly in need of the information. However, a lot of broadcast bandwidth is wasted when the demand for the information is low. Alternatively, the system may choose a pull-based information dissemination strategy, as shown in Figure 1(b). The soldier broadcasts a query for the information when it is needed. When the nodes that have the information receive query, they send the information to the soldier. This pull-based strategy is relatively more efficient than the push-based strategy when the frequency of query is relatively low compared to the frequency of the interested events, because communication takes place only when it is needed. A natural question is: can we combine the advantages of both push and pull strategies and build an efficient query-support mechanism that adapts to the frequencies of query and events?

In this paper, we propose a novel "comb-needle" query support model that integrates both push and pull data dissemination, and analyze its performance in large scale ad hoc wireless networks. Using this model, we are able to explore a whole spectrum of push and pull strategies shown in Fig.2. In the comb-needle model, each sensor node pushes its data to a certain neighborhood and the query is disseminated only to a subset of the network. More specifically, the query process builds a routing structure dynamically that resembles a comb and the sensor nodes push the data duplication structure like a needle, as illustrated in Figure 3. One can view this query process as combing for needles in the sands or haystack. The desirable properties of this mechanism include:

- In most cases, the comb-needle strategy is more effi-

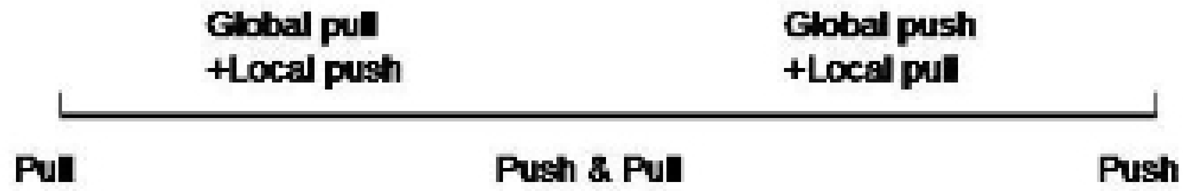

Figure 2: A Spectrum of Push-Pull Strategies

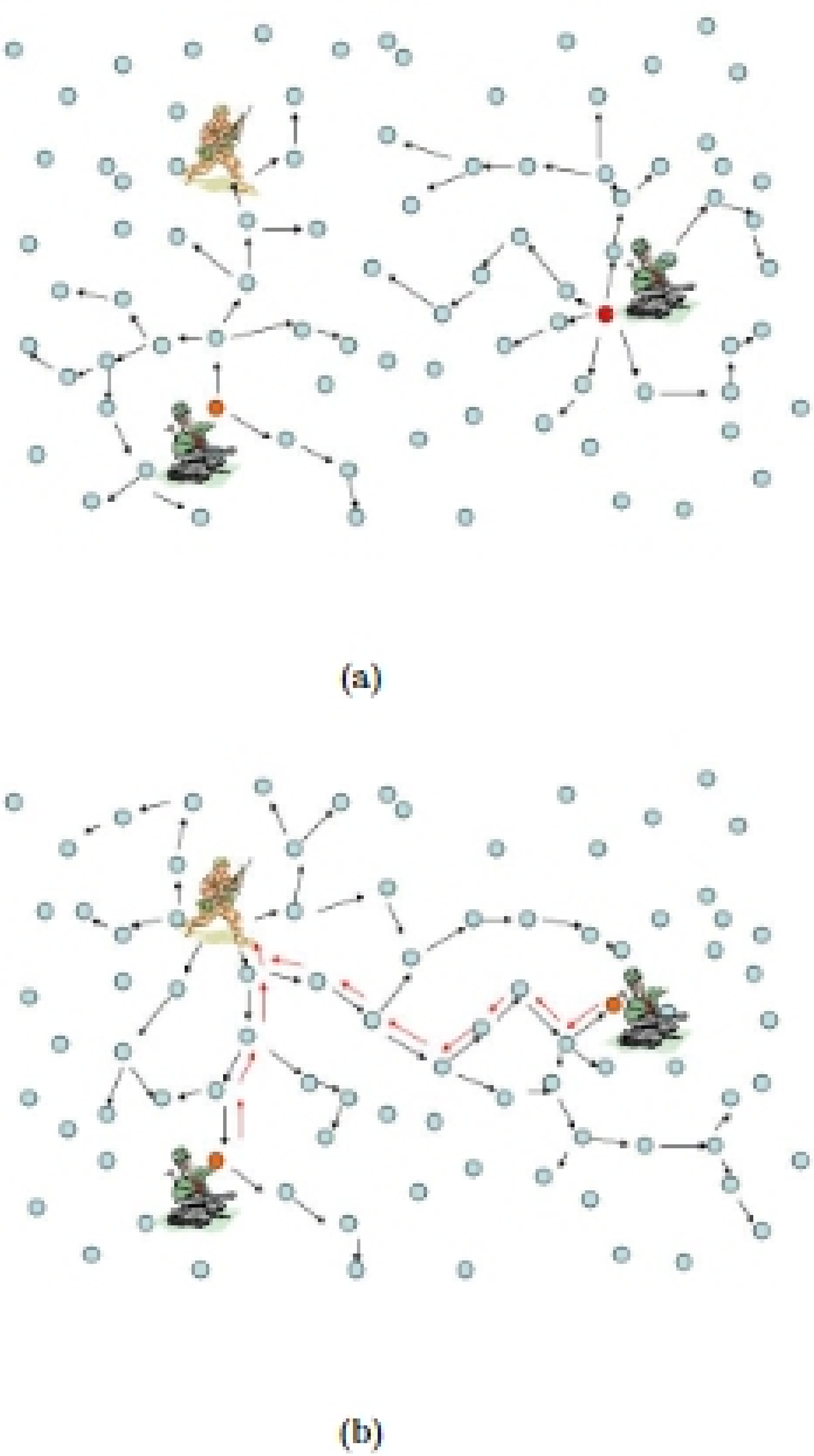

Figure 1: (a) Pure push-based strategy (b) Pure pull-based strategy

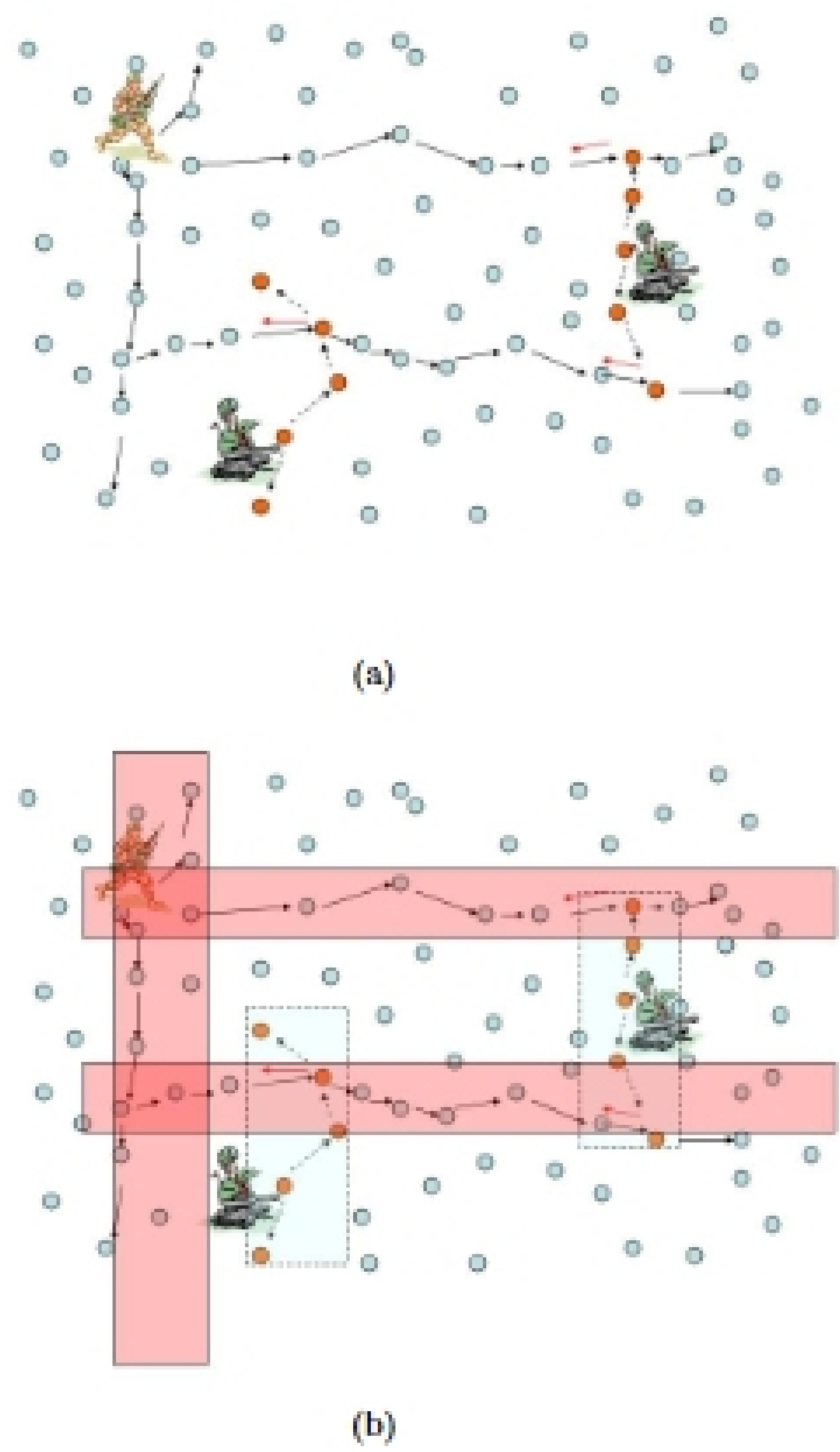

Figure 3: A Comb-Needle Example

cient than both pure push and pure pull strategies. Pure push and pull strategies can be consider as the end points of the combing strategy.

- The combing granularity is dynamically adjustable based on demand. No infrastructure is built *a priori*.

We analyze the communication cost of the comb-needle strategy based on query frequency and event frequency, explore the issue of query coverage and reliability in the case of faulty networks, and study adaptive strategies for the case where the frequencies of query and events are unknown *a priori* and time-varying.

This study adopts a few assumptions about the environment, the network, and the application scenario:

- the event of interest can occur randomly at anywhere at any time.

- the query entry point can be anywhere in the network and occur at any time.
- the network is ad hoc and uniform in the sense that all nodes are equivalent and the network does not have build-in hierarchy.
- all nodes in the network have information of their locations in the 2D space.

The remainder of the paper is organized as follows: in Section 2, we describe and analyze the comb-needle mechanism. In Section 3, we present simulation results for supporting queries on random grid networks. In Section 4, we explore the coverage issues in an unreliable network and present more robust strategies, which is followed by an adaptive comb-needle strategy in Section 5. Section 6 includes related work, which is followed by a discussion section and a conclusion section.

2. BALANCING PUSH AND PULL

2.1 Motivation

We consider the global discovery type query such as "where are all the tanks?", and "what locations have a temperature exceeding 90 degrees?" etc. For such queries, the whole network need to be traversed by the underlying query resolution mechanism in order to get a complete response. Note that not all queries require a global discovery. Queries like "are there 5 tanks in the region?" could only traverse part of the network (up to the point where the answer is positive.) Further, we assume that query entry can be any nodes in the network. Such query generation mode can support mobile information-gathering agencies (e.g., soldiers in battlefield) or hierarchical networks where higher hierarchies are more intelligent and may demand information.

For resolving a global query, a frequently used query dissemination mechanism is the flooding-based querying (FBQ)[9]. In FBQ, the underlying query support mechanism floods the request to the whole network. All nodes with relevant information respond. The flooding often represent the most significant message overhead for such queries. Note that we are only interested in one-shot query, rather than continuous query. In a continuous query, directed diffusion type of schemes work sufficiently well since the initial flooding cost can be amortized over the duration of the continuous information flow from the sensor nodes to the querier.

FBQ represents a pull-based query support mechanism. As noted in the introduction, there is a potential advantage in combining the push based strategy with the pull based strategy to increase query efficiency. For instance, let's consider a simple grid network like the one in Figure 4. Nodes $(4, 2)$ and $(7, 6)$ detected some abnormal event E, push the information vertically in the networks, and have it temporarily stored on the nodes traversed. Node $(0, 8)$ is an entry point of a discovery query interested in such events. Note that the query only needs to be disseminated horizontally to be resolved, i.e., the information pull only need to occur horizontally rather than throughout the network. Assume that the events are only detected once at the two nodes, and the query only occurred once as well. The total message cost for supporting the query is about 30 by this push-pull model, comparing the approximately 100 messages needed for the FBQ method. In the present hardware state, communication cost often weights much more than the storage cost, so this push-pull strategy seems desirable. One also can observe that the relative benefit of this push-pull strategy depends on the relative rates of occurrence of the events and query in the network. When there are more queries, there are more savings in the message overhead for supporting the queries. In this section we introduce a more general pull-push model and analyze its performance.

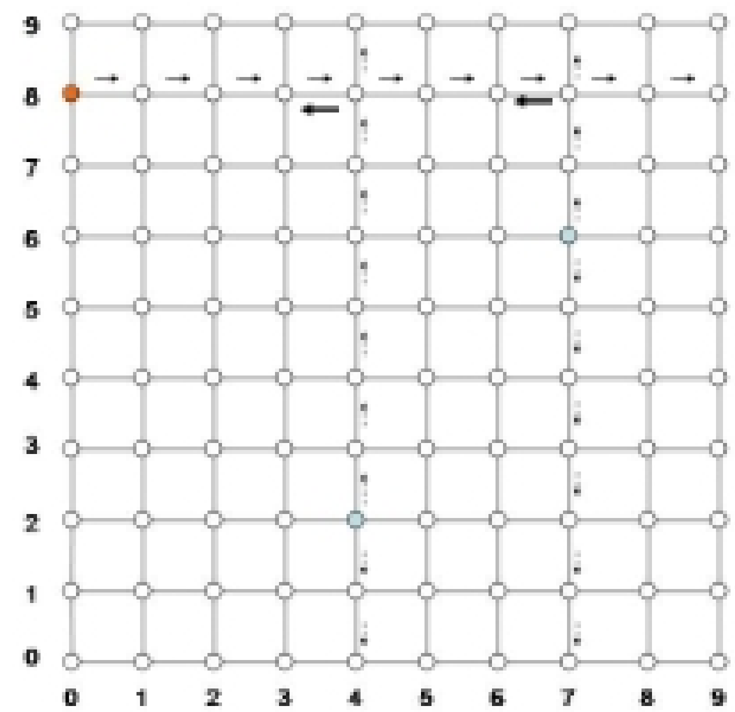

Figure 4: A Special Push-Pull Strategy

2.2 The Combing Strategy

The push-pull query support model we propose resembles the action of combing for needles in a haystack or in a pool of sand, and is thus dubbed as the "comb-needle" model. The comb-needle query support model combines both push and pull in the following way: each sensor node pushes its data to a certain neighborhood and the query node disseminates its request to a subset of the network. More specifically, the query process builds a routing structure dynamically that resembles a comb and the sensor nodes push the data duplication structure like a needle, as illustrated in Figure 3.

To analyze mathematically the cost and benefit of the query support mechanism, we need to have a model of the world that the sensor network is monitoring. In this paper, we assume the following random event and query models. Events occur uniformly in space and time across the sensor network. The discovery query entry point can be any node in the network with the same probability. Accordingly, we define two parameters:

f_q: the arrival frequency of discovery queries.

f_e: the arrival frequency of relevant events.

A life-time parameter could also be included in the query message to indicate whether it is interested in the events happened in the past half-an-hour or in the next five minutes. In this paper, we consider homogeneous query messages that are interested in events in one time unit.