

SDIP³: Structured and Dynamic Information Push and Pull Protocols for Distributed Sensor Networks

Ying Zhang and Qingfeng Huang

Palo Alto Research Center Inc.,
3333 Coyote Hill Rd,
Palo Alto, CA 94304, USA
{yzhang,qhuang}@parc.com

<http://www.parc.com/yzhang> and <http://www.parc.com/qhuang>

Abstract. We propose and study a class of structured and dynamic information push and pull protocols for wireless sensor networks. For structured information dissemination, our study focuses on the impact of various information demand characteristics on dissemination along some type of backbone structures. Our exploration of dynamic information push and pull focuses on finding optimal strategies in a distributed manner without prior knowledge of information demand characteristics and/or with heterogeneous query distributions. Our theoretical analysis uses a simple grid structure, but the protocol is applicable to arbitrary network topologies. A distributed traffic information system is used as the context of study and the simulation study uses a microscopic traffic simulator to demonstrate some of the ideas discussed in the paper.

Key words: Distributed algorithms for collaborative information processing, Communication and processing primitives

1 Introduction

The goal of this work is to advance the understanding of optimal information push and pull strategies for wireless sensor networks, with and without the knowledge about the spatial and temporal characteristics of information demand. Since data gathering and provision are sensor networks' main functions, efficient information gathering protocols and information dissemination/query/discovery protocols ([1] [2] [3] [4] [5]) are essential in building efficient sensor network applications. In this paper we focus on distributed information dissemination and discovery protocols.

This work is directly motivated by the following work and observations:

- Oftentimes query interest and frequency of an application exhibits location-varying characteristics. For instance, in an intelligent transportation system scenario, a vehicle could be more interested in traffic information on its potential paths to the destination and a piece of information about traffic

jam could be queried more frequently from those roads closer by than those further away.

- Previous work [2][6] have shown that knowledge about query frequency and data event frequency can be used to create efficient push-pull balanced information discovery and dissemination protocols. However, in these earlier work, query frequency is uniform or homogeneous across nodes/space;
- If the non-uniform spatial distribution of information demand is known, one maybe able to further optimize push-pull strategies;
- When good estimation about spatial distribution of query frequency does not exist a priori, information dissemination structure should try to adapt to ongoing event and query distribution patterns for improved efficiency.

As a result, our investigation starts from exploring how to use potential knowledge about spatial distribution of information demand, and investigating what to do when such information is not available.

The contributions of this paper include:

- A class of structured information push-pull protocols for applications that have spatially non-uniform information demand;
- Theoretical results for the performance of these protocols regarding optimal push-pull balance, given a variety of spatial distribution of query frequencies;
- A distributed dynamic push-and-pull protocol for scenarios where good estimation about spatial distribution of query frequency is not known a priori, and/or the query distribution is heterogeneous.

Comparing to related work in information dissemination area, we address the optimization perspective with the consideration of query frequency and distribution which is rather unique to our knowledge. Much related research on distributed query efficiency [6] [7] has been done in the last couple of years, including “comb-needle”[2], DIM [3], GHT [4] and DIMENSIONS [8]. However, all these related work only explicitly or implicitly focused spatially uniform queries. The same if true for another related line of work points to potential of reducing dissemination/query cost by taking into account application semantics [5, 9, 10] and the study of fundamental scaling laws on energy-efficient storage and querying such as [11]. Moreover, related work on tracking (STALK [12], LLS [13]) explored multi-resolution structures; and recent work on distance-sensitive information services ([14], Trail [15], Glance [16]) studied optimal structures for information dissemination. However, none of them has considered query distribution or frequencies either.

The rest of this paper is organized as follows. Section 2 provides the problem formulation. Section 3 presents some theoretical results on a grid structure. Section 4 describes the dynamic information push and pull protocols. Section 5 focuses on the traffic network application with simulation validation, followed by conclusions in Section 6.

2 Problem Formulation

Systematic analysis of the push-pull strategies relies on the underlying system, protocol, event generation and demand distribution models. In this section we define these models in our study.

2.1 System Model

We consider two topologies in our model of the distributed information system: a *information relevance network topology* and a *communication network topology*. In the example of distributed traffic information system, we represent each road segment as a node in the information relevance network, node X has a directed link to node Y if and only if the segment represented by Y is connected to the segment represented by X in the road traffic topology. Each node in the communication network represent a communication interface node and a link between two nodes represent their ability to communicate directly. Fig. 1 shows an example of a road configuration with traffic sensors distributed on road segments. Fig. 2 only shows one intersection in Fig. 1 with detailed road connectivity, which determines the connectivity in the corresponding information relevance network. $L(i, j)$ is a notation for a node in the information network, representing the road segment from point i to point j . $L(i, j) \rightarrow L(j, k)$ if there is legal turn at j .

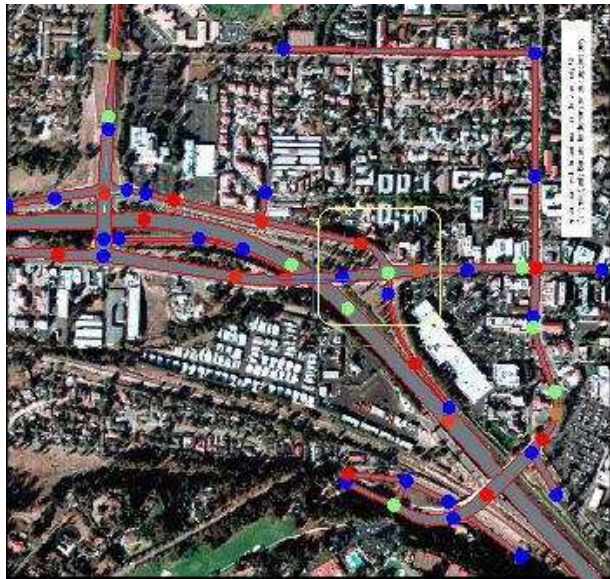


Fig. 1. A traffic information network, circles indicate locations of sensor nodes.

The example of communication network is shown on the right of Fig. 1 where an

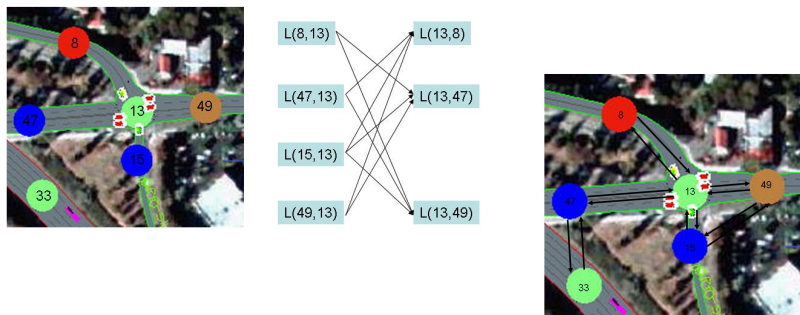


Fig. 2. Left: Traffic/road network abstraction; Right: Communication structure.

arrow from node i to node j indicates that node j can hear from node i . Note that although node 33 has no road connection to node 47, it can communicate with it.

For simplicity, in this paper we assume both the information network and the communication network are fixed.

2.2 Push and Pull Model

Our structured information dissemination strategy combines pro-active push and reactive pull. The push and pull model is formally defined as follows.

Push The structured information push model is specified by: $\langle m, R \rangle$ where topological constraint R defines the recipient set of the pro-active information push, m is a message that can be further defined as a 4-tuple: $m = \langle s_i, i, t, \tau \rangle$ where

- s_i is the ID of the information source,
- i is the information of that source,
- t is the time that the information is obtained,
- τ is the expiration time. When $\tau = \infty$, it is event driven, i.e., it will not be expired until the next publication.

One simple example constraint of R is the maximum “distance” the information should be pushed. Interestingly, for a traffic network, a meaningful distance may be represented by time, e.g., T seconds away (given current, historical or nominal vehicle speed information on relevant roads) to the source road segment.

More complicated constraint for R could represent fine grain topological structures that we may call “push backbone”. Fig. 3 is an example of such push

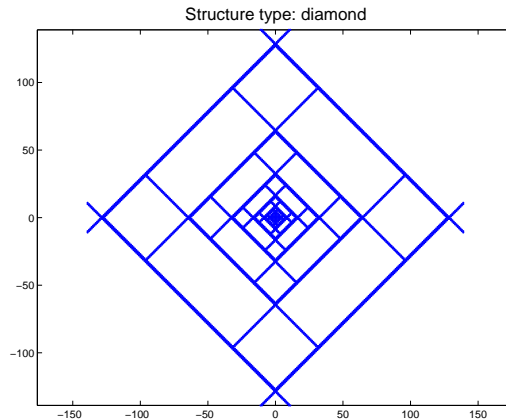


Fig. 3. A multi-resolution diamond structure

backbones. In this example, the source of information is located at the center, the structure is sparser when further away from the source. As one can see later, this kind of multi-resolution push structure would help increase the efficiency of information dissemination when information demand density have a distance varying distribution.

A push process can be carried out as follows. For simplicity in description, let us assume that the nodes within constraint R of a source are directly or transitively connected to the source. Each node use a list to keep track of the most recent updates it has heard about. When a node n receives a new piece of information $\langle s_i, i, t, \tau, R \rangle$, it checks its constraints in R : if R is not satisfied, or it is expired (i.e., $t + \tau$ is smaller than the current time), then the message is dropped, otherwise rebroadcast $\langle s_i, i, t, \tau, R \rangle$ and:

- if s_i does not exist in the list, add the entry with $\langle s_i, i, t, \tau \rangle$,
- if s_i exists in the list with smaller t , update the entry with $\langle s_i, i, t, \tau \rangle$.

Note that as we assume the recipient set R and the source are self-connected without other relay nodes, this simple implementation guarantees that all recipients within the expiration time bound will receive the information. If the communication links are not reliable, one can increase the thickness of the lines in the structure [2] to achieve the desired level of reliability. Note that in this implementation, nodes not in R may also overhear the information, in that case they shall stored the information in their lists as well if memory is less expensive than communication cost.

In our cost model, we assume each transmission costs one unit. When R is connected, the cost of dissemination is bounded by the number of nodes in R , i.e., each node in R transmits at most once. If a routing tree such as a connected minimum dominating set tree is built a priori, the number of transmissions can be less than the number of nodes in R , since leaf nodes do not need to transmit.

For simplicity in our analysis, we assume the cost of dissemination be the number of nodes in R .

Pull The query we are concerned with is specified by a triple: $\langle n_q, s_i, c \rangle$ where

- n_q is the ID of the node generating the query,
- s_i is the source of the information that is queried,
- c is a constraint on the information of interest on the respective source, (e.g, how “fresh” the information needs to be).

If we assume every node has the knowledge of the dissemination structure R , the most efficient way is to passing the query along the shortest hop-count to the structure. In this case, the cost of a query is twice of that hop-count since a query goes a round trip from the node to the structure. If the structure is not completely known, one may pass a query along a ”needle” that may hit the structure within certain hop-counts, and the cost of a query is proportional to the length of the needle. If the structure is completely unknown, a limited broadcast within certain distance is another choice. In this case, the cost of a query is the total number of nodes in the broadcast region. In this paper, we focus on the first case, however, similar analysis can be easily carried out for other cases.

The in-network query processing is carried out as follows. When a network node receives a query $\langle n, s_i, c \rangle$, it checks to see if it has relevant data about source s_i that satisfies the constraint c in the source list. If the information does not exist locally, the query will be forwarded towards the backbone. Along the way to its destination, as soon as the query hits a node that has the piece of information, a reply $\langle s_i, i, \min(\mathbf{t}, t + \tau), 0 \rangle$, where \mathbf{t} is the current time, is generated and propagated along the shortest path from this node to the origin of the query. Each node that receives or overhears the reply will update its source list with the information.

2.3 Event and Demand Distribution Models

We assume source s_i have event generation frequency f_i . The information demand frequency for s_i may be specified by a function $\mathbf{f}_q : X \rightarrow \mathcal{R}^+$, where X is a set of locations relative to the source of information s_i , \mathcal{R}^+ is the set of non-negative real numbers, and the value $\mathbf{f}_q(\bar{x})$ represents query frequency about information s_i from location \bar{x} . For simplicity, we start from a omni-directional version where query frequency is only related to the information distance from the query node to the source, i.e., $\mathbf{f}_q = \mathbf{f}_q(|\bar{x}|)$. We will use three types of frequency models:

- **uniform**: the frequencies at every location are a constant, i.e., $\mathbf{f}_q(d) = f_q$;
- **linear**: the frequencies are decreasing linearly with respect to the distance, i.e., $\mathbf{f}_q(d) = f_q(1 - kd)$ where $k > 0$ is a constant;
- **power**: the frequencies are inversely related to the distance, i.e., $\mathbf{f}_q(d) = f_q/d^\alpha$, where $\alpha > 0$ is a constant.

All three forms can be written as $\mathbf{f}_q(d) = f_q q(d)$ where f_q is a constant, and $q(d)$ is a function of d .

2.4 Formulation of the Optimization Problem

Our objective is, given the knowledge about the demand distribution and event generation frequency, find a push backbone structure R that minimize the overall communication cost. Let $C_{push}(R)$ be the cost of propagating one data packet from the source node on R ; let $C_{pull}(R, \bar{x})$ be the query cost for a node located at \bar{x} to find out the information from R . Note that this query cost depends on the push backbone structure R . The denser the push structure, the smaller the average query cost. With these definitions, the total push-pull cost rate (number of data packet per second) related to information s_i can be expressed as follows:

$$Cost(R) = f_i C_{push}(R) + \int_X C_{pull}(R, \bar{x}) \mathbf{f}_q(\bar{x}) d\bar{x} \quad (1)$$

The optimization problem can be formally expressed as: find an optimal push structure R_{opt} which minimizes the cost $Cost(R)$, i.e.

$$R_{opt} = \min arg_R Cost(R) \quad (2)$$

Since there is no obvious way to express in analytical form for the “structure” R , in this paper, we take the approach of exploring the optimal parameters within a specific class of structures.

3 Theoretical Analysis

In this section, we will do some preliminary analysis of the structures described in the previous section, and find out the closed-form solutions of the optimal parameters for the structures, with respect to three different demand distribution models: **uniform**, **linear** and **power**, using the following two structures. Let $\bar{x} = (x, y)$:

- **disk**: the only constraint of this structure is the push scope R_i , i.e., $d \leq R_i$ where $d = |x| + |y|$ for a grid network;
- **diamond** (Fig. 3): let r be the size of the base diamond, $b_+ = \text{sign}(x + y) \lceil r^{\lceil \log_r |x+y| \rceil} \rceil$, $b_- = \text{sign}(x - y) \lceil r^{\lceil \log_r |x-y| \rceil} \rceil$, $c_+ = \lceil r^{\lceil \log_r |x+y| \rceil + 1} \rceil$, and $c_- = \lceil r^{\lceil \log_r |x-y| \rceil + 1} \rceil$, the diamond structure can be represented by:

$$|b_+ - (x + y)| \leq 1 \wedge |x + y| \leq c_+ \vee |b_- - (x - y)| \leq 1 \wedge |x - y| \leq c_-.$$

Let the information generation and query frequencies be f_i and \mathbf{f}_q , respectively. Let $\mathbf{f}_q(d) = f_q q(d)$. Let R_i and R_q be the push and pull scopes, respectively. C_{push} is equal to the total number of nodes in R . $C_{pull}(R, \bar{x})$ is the number of transmissions of a query at \bar{x} , which is twice of the distance from \bar{x} to the structure R . Given R is omni-directional, we have $C_{pull}(R, \bar{x}) = C_{pull}(d)$ where

$d = |\bar{x}|$. The total cost for information push is $C_{push} = f_i C_{push}$ and the total cost of information pull is $C_{pull} = f_q C_p$ where $C_p = \sum_{d=0}^{R_q} q(d) C_{pull}(d) 4d$. Here $4d$ is the number of nodes at distance d for a grid network. For each case, we estimate C_{push} and C_{pull} and then find the optimal parameter by solving equation $\frac{\partial(C_{pull}+C_{push})}{\partial p} = 0$ where p is a parameter of the structure R .

3.1 Optimal Push Scopes for Disk Structures

For a disk structure with push scope R_i , it is easy to see that the total number of nodes in R is about $C_{push} = 2R_i^2$, and $C_{pull}(d) = 0$ if $d \leq R_i$ and $C_{pull}(d) = 2(d - R_i)$ otherwise, where the factor 2 comes from the fact that query is a process involving a round trip.

Uniform ($q(d) = 1$)

$$C_p = \sum_{d=R_i}^{R_q} 2(d - R_i)4d = 8 \sum_{d=R_i}^{R_q} d(d - R_i) \sim 8((R_q^3 - R_i^3)/3 - R_i(R_q^2 - R_i^2)/2)$$

Let $\frac{\partial(C_{pull}+C_{push})}{\partial R_i} = 0$, we have an optimal push and pull balance scope condition, by $f_i R_i = f_q(R_q^2 - R_i^2)$ We can see that if f_q or f_q/f_i approaches 0, R_i should approach 0, i.e., no push is needed. On the other hand, if f_i or f_i/f_q approaches 0, $R_i \rightarrow R_q$, i.e., push as far as the query scope.

Linear ($q(d) = 1 - d/R_q$)

$$C_p = \sum_{d=R_i}^{R_q} 2(d - R_i)4d(1 - d/R_q) = 8 \sum_{d=R_i}^{R_q} d(d - R_i) - \sum_{d=R_i}^{R_q} d^2(d - R_i)/R_q \\ \sim 8((R_q^3 - R_i^3)/3 - R_i(R_q^2 - R_i^2)/2) - 8((R_q^4 - R_i^4)/4 - R_i(R_q^3 - R_i^3)/3)/R_q$$

Let $\frac{\partial(C_{pull}+C_{push})}{\partial R_i} = 0$, we have an optimal push and pull balance scope condition, by $f_i R_i = f_q(R_q^2 - R_i^2) - \frac{2f_q}{3R_q}(R_q^3 - R_i^3)$ It is easy to see that $R_i^{linear} < R_i^{uniform}$.

Power ($q(d) = 1/d$)

$$C_p = \sum_{d=R_i}^{R_q} 2(d - R_i)4d/d = 8 \sum_{d=R_i}^{R_q} (d - R_i) \sim 8((R_q^2 + R_i^2)/2 - R_i R_q)$$

Let $\frac{\partial(C_{pull}+C_{push})}{\partial R_i} = 0$, we have an optimal push and pull balance scope condition, by $f_i R_i = 2f_q(R_q - R_i)$ i.e. $R_i = \frac{R_q}{1+f_i/(2f_q)}$. It is easy to see that $R_i^{power} < R_i^{uniform}$.

3.2 Optimal Diamond Size

In this case, we assume the push scope is as far as the query scope, i.e., $R = R_i = R_q$, and find out the optimal spacing parameter given a diamond structure (Fig. 3). Note that the line thickness can vary; in most cases, it has to be at least 2 for guarantee the connectivity of the structure. In the analysis below, we use β for the line thickness in the structure. Given the the base diamond size r , the cost of push for each dissemination is

$$C_{push} = \beta \sum_{k=0}^{\lfloor \log_r R \rfloor} 4r^k \sim \beta 4R/(r-1)$$

Here we use $r^{\lfloor \log_r R \rfloor} \sim R$ to simplify the analysis. The spacing between two lines at distance d is $r^{\lfloor \log_r d \rfloor + 1} - r^{\lfloor \log_r d \rfloor} \sim (r-1)d$. The average pull cost for a query $C_{pull}(d)$ is about $2(L/4) = L/2$ where $L = (r-1)d$. Note that for this structure the pull cost is increasing with respect to the distance.

Uniform ($q(d) = 1$)

$$C_p = \sum_{d=0}^{d=R} 4d(r-1)d/2 \sim \frac{2}{3}R^3(r-1)$$

$\frac{\partial(C_{pull}+C_{push})}{\partial(r-1)} = 0$, yields an optimal diamond size, by $6\beta f_i = f_q R^2(r-1)^2$, i.e.
 $r-1 = \frac{\sqrt{6\beta f_i/f_q}}{R}$.

Linear ($q(d) = 1 - d/R$)

$$C_p = \sum_{d=0}^{d=R} 2d^2(r-1)(1-d/R) \sim \frac{2}{3}R^3(r-1) - \frac{1}{2}R^3(r-1) \sim \frac{1}{6}R^3(r-1)$$

$\frac{\partial(C_{pull}+C_{push})}{\partial(r-1)} = 0$, yields $24\beta f_i = f_q R^2(r-1)^2$, i.e., $r-1 = 2\frac{\sqrt{6\beta f_i/f_q}}{R}$.

Power ($q(d) = 1/d$)

$$C_p = \sum_{d=0}^{d=R} 2d^2(r-1)/d \sim R^2(r-1)$$

$\frac{\partial(C_{pull}+C_{push})}{\partial(r-1)} = 0$, yields $4\beta f_i = f_q R(r-1)^2$, i.e., $r-1 = 2\sqrt{\beta \frac{f_i/f_q}{R}}$.

It is easy to see that $r^{linear} \sim 2r^{uniform}$ and $r^{power} \sim \sqrt{\frac{R}{6}}r^{linear}$.

3.3 Push Scope Verification

We use *load factor* as a performance metric, which is the total amount of pushes and pulls for a piece of information, normalized by $n(f_q + f_i)$ where n is the number of nodes involved.

In the study, we assume the pull scope to be 150 hop-count to a source in a grid network. Let $f_i = 1$ without loss of generality, and set f_q 0.01 and 0.05 for **uniform**, and 0.05 and 0.1 for **linear**, and 0.1 and 0.2 for **power** models, respectively.

The load factor is computed using Eq. 1, where $C_{push}(R)$ is the number of nodes in R and $C_{pull}(R, \bar{x})$ is $2h$ where h is the minimum hop-count from \bar{x} to structure R .

Fig. 4 show the load factors using the **disk** structure, for **uniform** and **power** query models, respectively.

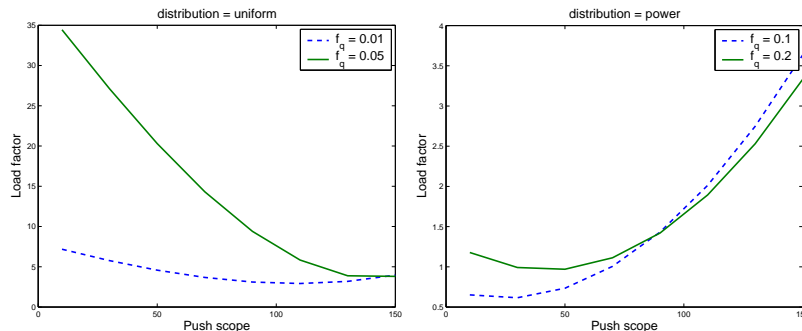


Fig. 4. Load factors for (Left) the uniform and (Right) the power query models

From the theoretical analysis, we know $f_i R_i = f_q (R_q^2 - R_i^2)$ for the uniform query model. Given $f_i = 1$ and $f_q = 0.05$, and $R_q = 150$, we obtain $R_i = 108$ which is consistent to the left curve in Fig. 4. Similarly, for $f_q = 0.2$, we have optimal $R_i = R_q / (1 + f_i / (2f_q)) = 43$ for the power model, which is consistent to the right curve in Fig. 4.

4 Dynamic Balancing of Push and Pull

In the previous section, we see that the average-case optimal values of a information push structure depends on two key factors: the rate of information generation and the rate of request for the information. Once one collects the historical information regarding these two rates in the information network, one can determine the average-case optimal push parameters for expected rates of information generation and request.

This average-case optimality can be further improved by capturing and responding to the fluctuations in the information generation and request rates in

real-time. We propose a distributed dynamic optimization scheme called *micro-balancing* to achieve this goal. Given the recipient set R , the micro-balancing scheme reactively changes R to minimize the total cost of information sharing. The rest of this section describes this mechanism. Such a scheme can be also applied to cases where the query distribution is not omni-directional or unknown at all.

4.1 Boundary states

We use two boolean values to denote the boundary of push dissemination. Assume the push structure forms a spanning tree rooted at the source. A node in the structure is in state 1 if it needs to forward the information and in state 0 if it does not need to forward the information. The set of nodes in state 1 is called *active* nodes. A node in the structure is in boundary state 0 if it is in 0 but its parent is 1. A node in the structure is in boundary state 1 if it is in 1 but all its children in the structure are in 0.

For micro-balancing, the push boundary can be changed dynamically according to the push and pull rates. To obtain the push and pull rates, nodes in both boundary states count the number of push events and the number of queries. Let N_q be the number of different queries a node receives, and N_p be the number of different “pushed” information a node receives. It is called *information-dominate* if $\frac{N_p}{N_q} > 2 + \sigma$ and *query-dominate* if $\frac{N_p}{N_q} < 2 - \sigma$ where σ is a small value parameter for the purpose of reducing the oscillation in state-change condition boundary to reduce potential protocol overhead. The insight that leads to the inequality is a simple one: a query and a reply need two uses of the wireless medium and a push only need one local broadcast. To make the switch between these two conditions more stable for small N_q and N_p , we compute the inequalities only after $N_q + N_p > \Delta$ for some constant Δ .

4.2 Boundary state transition

The rules for state transition are as follows:

- 0 \rightarrow 1: if it becomes query-dominate,
- 1 \rightarrow 0: if it becomes information-dominate

Whenever a node in the boundary state changes its state, its neighbor’s state will be changed. For example, if a node changes to state 1, all its children become boundary state 0; if a node changes to 0, its parent may be a boundary node 1. When a node enters the boundary state, both N_p and N_q are reset to 0. It is easy to see that the state transition guarantees that the set of active nodes are connected.

Note that micro-balancing will not change the push structure, but it changes the active nodes in the structure, forming a more efficient push structure that is optimal for an unpredictable query model.

The push operation is the same as before except each node at the structure need to check if it is active or not. Only active nodes participate in the push.

5 Application: Distributed Traffic Information Networks

5.1 Traffic Network Simulations

We have implemented and simulated the push-pull protocols using a commercial on-the-shelf microscopic traffic simulator Paramics [17]. Fig. 1 shows a traffic network in Paramics Simulator (demo5 in Paramics software distribution) that we use for testing our protocol. The area of the network is $1466.6\text{m} \times 1102.3\text{m}$. This network has 58 intersections and 118 road segments. Our protocol and event/query generations model are implemented as a plug-in to the simulator. We assume each road segment generate some events with probability p_e at every second, and each car at a road segment with a probability p_q queries another road segment that it may travel within a time bound T_q . Parameters p_e and p_q are used to control the event and query frequencies. The communication range is set to be 300m. The simulation data are collected within a time window $[10s, 1010s]$. We record the total number of transmissions for pull and push at each node. The performance metrics is the total number of transmissions, i.e., the sum of pushes and pulls over the whole network.

A shortest path routing table is built up at the initialization stage for query propagation. For the static push protocol, any piece of new information will travel up to nodes that are within T_i seconds away from the source along the road network, where T_i is a parameter set at initialization. For the dynamic push, we set σ be 0.1 and Δ be 5. Since the network is small, we set $T_q = 100$ which effectively includes all the links when querying at any node. We also assume information to be event-driven, i.e., $\tau = \infty$. And query constraint c is a timeout that equals to the travel time to the location of the information. Note that in this simulation, we did not use linear or power query models since the size of the traffic network is too small. A larger size simulation will be conducted in the future.

Our implementation also includes visualization of communication. In Fig. 1, for example, colors of circles indicate push/pull ratios, where red is pull dominated and blue is push dominated.

5.2 Simulation Results

The purpose of the simulation is to verify our theoretical analysis about push and pull scope, and show the effectiveness of the dynamic push and pull protocol. Three cases are studied:

- I. Information dominate ($p_e = 0.01, p_q = 0.03$): in this case push scope should be small;
- II. Query dominate ($p_e = 0.001, p_q = 0.3$): in this case push scope should be as large as pull scope;
- III. Neither ($p_e = 0.01, p_q = 0.3$): in this case push scope should be somewhere in the middle.

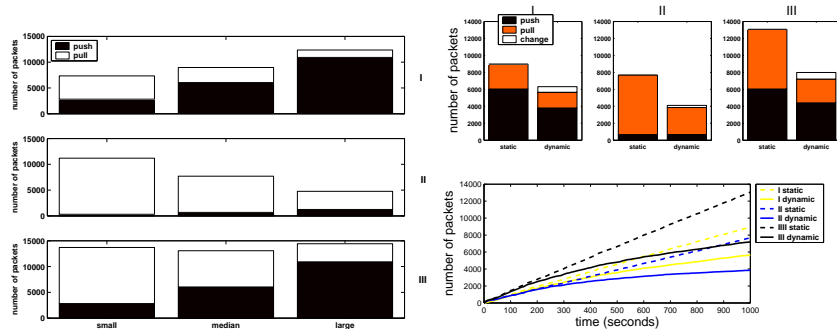


Fig. 5. Left: Number of transmissions for three cases and three push scope settings, Right: Static vs. dynamic push strategies.

First we test the optimal scopes for the static push protocol. We set the push scope to be small $T_i = 5$, median $T_i = 15$, and large $T_i = 30$, for each case. A total of nine sets of data are collected and compared. Figure 5 Left shows the results. We observe that number of pushes increases and number of pulls decreases with the increase of the push scope, and for case I, push scope small is most effective, for case II, push scope large is most effective, and for case III the ideal push scope is somewhere in the middle.

We also tested the effectiveness of the dynamic push protocol. We set the initial push scope $T_i = 15$ for each case, then collect data for both static and dynamic push strategies. A total of six sets of data are collected and compared. Figure 5 Right shows the results. Note that in addition to pushes and pulls, the dynamic protocol introduces the extra state change cost, each of which costs one transmission.

We observe that in all these cases the dynamic protocol reduces the total cost by about 50%. There are two reasons that the dynamic protocol outperforms the static one: (1) when the pull frequencies are unknown, the dynamic protocol can adjust to the push scope to balance the demands and changes, and (2) when the pull frequencies are not uniformly distributed over all directions, the dynamic protocol can adjust the push scope locally at different directions.

6 Conclusions

In this paper we proposed a class of the structured information push and pull protocols. We first analyzed these protocols using various query models and then presented dynamic micro-balancing strategies when the query models are unknown. We also applied this class of protocols to the distributed traffic information system and demonstrated the benefit of push-pull micro-balancing.

References

1. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks."
2. X. Liu, Q. Huang, and Y. Zhang, "Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 122–133.
3. X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *Proceedings of the first international conference on Embedded networked sensor systems*. ACM Press, 2003, pp. 63–75.
4. S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensor networks with GHT, a geographic hash table," *Mob. Netw. Appl.*, vol. 8, no. 4, pp. 427–442, 2003.
5. B. Krishnamachari and J. Heidemann, "Application-specific modelling of information routing in sensor networks," in *Proceedings of the IEEE International on Performance, Computing, and Communications Conference*. Phoenix, Arizona, USA: IEEE, April 2004, pp. 717–722.
6. S. Kapadia and B. Krishnamachari, "Comparative analysis of push-pull query strategies for wireless sensor networks."
7. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM Press, 2003, pp. 491–502.
8. D. Ganesan, D. Estrin, and J. Heidemann, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?" in *Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I)*, Princeton, NJ, October 2002.
9. J. Heidemann, F. Silva, and D. Estrin, "Matching data dissemination algorithms to application requirements," in *Proceedings of the first international conference on Embedded networked sensor systems*. ACM Press, 2003, pp. 218–229.
10. N. Sadagopan, B. Krishnamachari, and A. Helmy, "Active query forwarding in sensor networks," *Elsevier Journal of Ad Hoc Networks*, 2003.
11. J. Ahn and B. Krishnamachari, "Fundamental scaling laws for energy-efficient storage and querying in wireless sensor networks," in *Proceedings of International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006.
12. M. Demirbas, A. Arora, T. Nolte, and Lynch, "A hierarchy-based fault-local stabilizing algorithm for tracking in sensor networks," in *International Conference on Principles of Distributed Systems*, 2004.
13. I. Abraham, D. Dolev, and D. Malkhi, "Lls: a locality aware location service for mobile ad hoc networks," in *Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications*, 2004.
14. S. Funke, L. J. Guibas, A. Nguyen, and Y. Wang, "Distance-sensitive routing and information brokerage in sensor networks," in *IEEE International Conference on Distributed Computing in Sensor System (DCOSS'06)*, June 2006, pp. 234–251.
15. V. Kulathumani, A. Arora, M. Demirbas, and M. Sridharan, "Trail: A distance sensitive network protocol for distributed object tracking," in *European conference on Wireless Sensor Networks (EWSN)*, 2007.
16. M. Demirbas, A. Arora, and V. Kulathumani, "Glance: A lightweight querying service for wireless sensor networks," in *International Conference on Principles of Distributed Systems*, 2006.
17. "Paramics," website. [Online]. Available: <http://www.paramics-online.com/>