

Application-Specific Modelling of Information Routing in Wireless Sensor Networks*

ISI-TR-576 August 2003

Bhaskar Krishnamachari[†] and John Heidemann[§]

[†]Department of Electrical Engineering

[§]Information Sciences Institute

University of Southern California

bkrishna@usc.edu, johnh@isi.edu

Abstract

Sensor network applications have a diverse set of requirements—some involve extraction of sensor data to a single point, others exploit sensor-to-sensor communication; some employ long-lasting data streams while connections in others are mainly ephemeral. Different variants of the directed diffusion routing protocol—pull-based, push-based and hybrid rendezvous-based—have been developed, along with in-network processing and geographic routing techniques. However, there has been no prior systematic study comparing their performance with respect to the diverse application characteristics. In this paper, we develop novel abstract parameterized models for traffic and topology that can incorporate data aggregation and geographic scoping. Using these models, we mathematically analyze the performance of these routing techniques across a range of application scenarios (with varying numbers of nodes, sources, sinks, data settings etc.). Besides quantifying the conditions under which the different routing algorithms outperform each other, we obtain a number of useful design insights. Our analysis shows that algorithms mismatched to applications can result in drastically poor performance; demonstrates the desirability of reducing flooded interest and exploratory messages when data aggregation is used; and suggests that it may be difficult to implement efficient hybrid schemes because their performance is very sensitive to the optimal placement of rendezvous points.

1 Introduction

Routing in multicast ad hoc networks, and now sensor networks, has been a rich area of research. Work in mo-

bile ad hoc networks explored pro-active protocols such as DSDV and TORA, reactive protocols such as AODV and DSR, and hybrid approaches such as ZRP [15]. In sensor networks as well, a wide range of data dissemination protocols have been explored, from protocols supporting attribute-addressed, peer communication (such as directed diffusion [9]) to protocols supporting tree-based communication to a single sink with an SQL abstraction [14]. Optimizations now being considered include geographic routing [11, 18], multipath routing (for example, see [1]), rendezvous-based approaches [3, 16], and many others.

Above routing, an increasing range of sensor networks applications are being developed. A common class of applications uses a sensor net to communicate data from the net to a single sink, possibly with opportunistic data processing along the way [9, 14]. More complex applications do controlled in-network processing, for purposes such as collaborative signal processing [17], to localize computation with nested queries [8], or to place database operators [2, 14]. Other applications involve increasingly sophisticated operation, with multiple kinds of distributed interaction and communication, including point-to-point state transfer and region-based suppression [13].

As the choices of protocols and the sophistication of applications grow, an important problem is the selection of the routing algorithm best matched to a given application. This problem is particularly important in sensor networks where limited resources force exploitation of application constraints [6]. To this end, we have developed several versions of directed diffusion [9, 18, 7]. The original directed diffusion algorithm employed flooding of data interests [9]. While appropriate to applications with a single data sink, its overhead increases when many nodes become interested in data. We have explored augmenting this mechanism with geographic scoping [18], and explored making data seek

*This work was supported by DARPA under grant DABT63-99-1-0011 as part of the SCADDS project.

interested sinks rather than the opposite [7]. Other possible approaches include hybrid rendezvous-based techniques [3, 16].

Experiments with a mix of applications and protocols show that the choice of protocol can make a large difference in performance, with the overhead reduced by 40–60% when dissemination protocols are selected with the application in mind [7]. But we have found that it is difficult, in practice, for application designers to select which algorithms to employ. There are several possible reasons for this problem. Application designers are often experts in their problem domain, and should not be expected to know the details of routing. Also, brief descriptions of the algorithms may not make it clear which algorithm best matches a given application’s needs. In fact, our experience has been that it is even easy to be misled, and find that what seems like a better match may be worse because of overheads in message size or route maintenance. Finally, although there is some analysis of basic diffusion [12, 10], there has been no prior systematic analysis of the relative performance of different versions of diffusion over different application scenarios.

The contribution of this paper is to help answer the question of how well diffusion routing algorithms match different applications. We develop suitable abstract models for application topology as well as application traffic, and an analytic framework for four variants of attribute-based routing in sensor networks. We then evaluate the performance of these routing techniques on different application scenarios, varying the topology and the number of involved sources and sinks.

Our results quantify the intuition that push-based diffusion performs better in terms of overhead than pull-based when there are many active data sinks and few data active sources (and *vice versa*). We are able to show the relative performance of these variants mathematically under different sets of assumptions such as the availability of geographic information and data aggregation. Our results also demonstrate the importance of matching the choice of routing to the application and provide a number of useful design insights.

The rest of the paper is organized as follows. In section 2, we briefly describe the different diffusion mechanisms. We describe the abstract models for the routing schemes, the topology and the data traffic in 3. The details of the mathematical performance analysis and results are provided in section 4. We briefly discuss related work in section 5 and provide concluding comments in section 6.

2 Description of Diffusion Mechanisms

In the abstract, one can consider sensor networks as distributed event-based systems. In these systems, *sources* generate or *publish* information observed from their environment; *sinks*, in turn, *subscribe* to this information. The role of data dissemination algorithms is to move data from sources to sinks efficiently, allowing applications to process the data in-network.

The process of matching sources and sinks can be done with several different algorithms. We consider three alternatives — pull-based diffusion (two-phase and one-phase), push-based diffusion, and hybrid rendezvous-based approaches. We will describe each approach in detail in the following subsections.

2.1 Two-phase pull diffusion

Initial work with diffusion used an algorithm we now call *two-phase pull* [9]. Sinks identify data by a set of attributes and this information propagates in an *interest* message that is by default flooded through the network (unless optimizations such as geographic scoping are available). Interests establish *gradients* as they flow through the network.

When an interest arrives at a data producer, that source begins sending data. The first data message sent from the source is marked as *exploratory* and is sent to all neighbors that have matching gradients. Although the scope of exploratory data can be limited as with interests, by default it is flooded to all nodes. When exploratory data arrives at the sink, the sink *reinforces* a neighbor (typically based on lowest latency), establishing a *reinforced gradient* towards itself. Reinforcements propagate hop-by-hop back to the sources, generating a tree of reinforced gradients from all sinks to sources. Following data messages are not marked exploratory, and are sent only on reinforced gradients rather than to all neighbors.

Nodes can also generate *negative reinforcements* if they receive data that is not relevant to them. Negative reinforcements are usually used to correct incorrect routes, such as after a topology change. Gradients are managed as soft-state, so that both interests and exploratory data occur periodically to refresh this state. Interests are sent every *interest interval*, exploratory data every *exploratory interval*.

2.2 One-phase pull diffusion

We have refined two-phase pull diffusion to eliminate one of the search phases. The revised algorithm is called

one-phase pull. As with two-phase pull, subscribers send interest messages that disseminate through the network, establishing gradients. In one-phase pull the sources do not send exploratory data, but instead send data to only the lowest-latency gradient corresponding to each sink. To distinguish between multiple sinks, each interest carries a flow ID. Flow IDs must be unique (either via MAC-addresses or probabilistic approaches), and thus one-phase pull does not have the strictly localized nature of two-phase pull. However, elimination of exploratory data can greatly reduce control message overhead in one-phase pull.

2.3 Push diffusion

Complementing pull diffusion, push diffusion makes the sources the active parties. The application uses the same interface as two-phase diffusion (except for a flag to indicate “push”), but underneath the implementation, the roles of the source and sink are reversed. Sinks are passive, with interest information remaining local to the node subscribing to data. Sources become active: when they generate data, they send exploratory data throughout the entire network (or to areas limited by geographic or prior information, if available). As with two-phase pull, when exploratory data arrives at a sink, a reinforcement message is generated and it recursively passes back to the source creating a reinforced gradient. Non-exploratory data follows only these reinforced gradients. Push can also take advantage of GEAR-style geographic optimizations [18].

Push was inspired by applications that have many sinks, but where there are only a few active sources. It was inspired by discussions with researchers at Sensoria, University of Wisconsin, and PARC who encountered applications with these requirements. Our expectation is that one-phase pull will work well for applications with many sources, and push for applications with many sinks. However, this has not been systematically explored before.

2.4 Hybrid, rendezvous approaches

To balance the costs of push and pull, one could have both sources and sinks meet at a pre-defined rendezvous point. In general, the sources and sinks both send information about interest or presence of data to the rendezvous point and if both are present a path is created.

The challenge with such hybrid approaches is the location of the rendezvous point. For sensor networks where data is sent to a single node (such as a user node, or an connection to the Internet), that node is the best choice since all data must be sent there. However, when there is no pre-determined extraction point, nodes must

still agree on a rendezvous scheme. Concentration of all traffic through a single node faces bandwidth limitations and will disproportionately consume energy at nodes near the extraction point. Several alternatives exist for sensor networks. Geographic Hash Trees map each data item to a point in space [16], thus evenly distributing load at the cost of locality. In Rumor Routing, sources send data along one line, sinks on another, non-parallel line, and their intersection forms the rendezvous point [3].

In each of these cases, passing data through the rendezvous point (RP) introduces overhead, since it may not be located on the optimal path between the sources and sinks. In this paper we consider the best-case performance of hybrid rendezvous protocols where the RP is located directly between the source and sink. However, as we shall see, when the RP location is not on the shortest path, significant additional overhead is introduced.

2.5 Geographic Information and Data Aggregation

In addition to these basic approaches, the physical nature of most sensor networks allows geographic information to be used to constrain search. GEAR (Geographic and Energy-Aware Routing) extends diffusion when node locations and geographic queries are present [18]. Although originally designed for pull diffusion, it has also been applied to push.

Finally, data aggregation is an important part of making sensor network communication efficient [9, 12]. Exactly how aggregation proceeds is application specific. In the best case (for example, if the user’s request is to find the maximum temperature in a region, or if duplicate readings can be simply suppressed), n messages about the same event can be replaced with one. We will analyze diffusion performance both with and without such best-case aggregation to establish bounds on performance.

3 Description of Model

In mathematically analyzing the performance of these routing mechanisms, the principal challenge lies in constructing an abstract model that is analytically tractable but also captures important aspects of a realistic scenario.

In addition to varying the algorithm, we wish to consider scenarios with and without data aggregation and with and without geographically directed queries/interests. Also, in our modelling, we seek to include topological considerations such as the number of sources, sinks, distances (in hops) between pertinent

	No Aggregation	Data Aggregation
Flooding	<u>NAF</u> • Pull (2-phase) • Pull (1-phase) • Push	<u>AF</u> • Pull (2-phase) • Pull (1-phase) • Push
Directed (Geographic)	<u>NAD</u> • Pull (1-phase) • Push • Rendezvous	<u>AD</u> • Pull (1-phase) • Push • Rendezvous

Figure 1: Classification of scenarios with respect to availability of data aggregation and geographic information; this figure also shows the mechanisms that are analyzed for each scenario.

nodes, and also application-specific characteristics such as the rate of event and interest generation.

3.1 Routing Assumptions

The routing costs and overheads in sensor networks can depend significantly on some underlying assumptions about the routing protocol—in particular, upon the availability of geographic routing information and the availability of data aggregation. In general, all interests and event notifications are assumed to be flooded within the network. However, if geographic information is available, it is possible to reduce the setup costs by directing such interests/event notifications only to intended recipients. Data routing costs can also be reduced by aggregating information from multiple data sources in-network. Figure 1 shows the four scenarios (labelled NAF, NAD, AF, AD) we consider in our modelling, and the mechanisms that we will analyze for each scenario.

3.2 Data Traffic Model

We will consider a simple data traffic model. Time is broken into distinct epochs. Each epoch is divided into two phases: setup and data-flow, as illustrated in figure 2. Although this model mimics the mechanics of the diffusion protocol’s interest messages, logically it can also be thought of as representing an abstract amount of data sent over an arbitrary period.

There are I data sources and J sinks. In the setup phase, each of the I data sources generates a new event independently with probability l_i and each of the J data

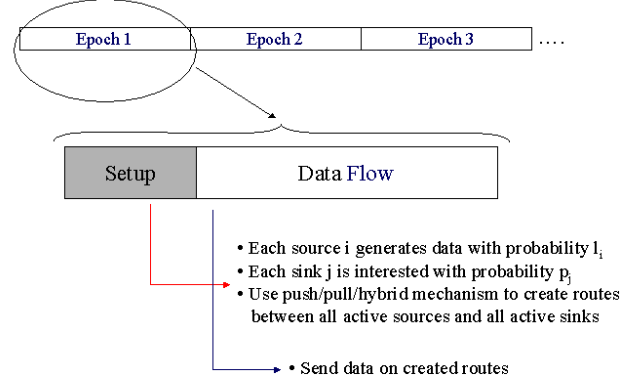


Figure 2: Traffic model: repeated epochs divided into setup and data-flow phases

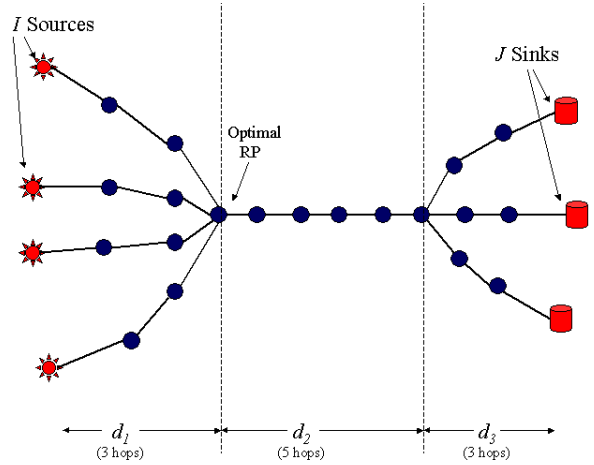


Figure 3: Illustration of source-sink topology

sinks independently generates an interest in the data (from all sources) with probability p_j . Depending on the mechanism being analyzed (push, pull, or rendezvous), these interests and events are notified to the relevant sources and sinks, and the pertinent routes are established in the setup phase. In the data flow phase, these routes are then utilized to sent information from generating sources to all interested sinks.

3.3 Topology Model

The overhead and performance of publish-subscribe mechanisms in sensor networks are impacted by the specific locations of the sources and sinks and the routes that the interest/event notifications and data packets follow. For the purpose of tractable and systematic analysis, we need an abstract and simplified model that captures these characteristics.

Besides the number of sources and sinks, since we con-

	S_I	$S_E = (\alpha S_D + S_C)$	S_R	S_D
two-phase pull	2	4.1 = (4 + 0.1)	4	1800
one-phase pull	2	—	—	1800
push	—	4.1	4	1800
rendezvous	2	4.1	4	1800

Table 1: Typical relative rates of control and data traffic (in Bytes/second).

sider models involving aggregation, the structure of the aggregation tree and the distance (hops) that aggregated data is carried within the network also influence performance. We therefore use the abstract topology illustrated in figure 3. The data from all active sources (i.e. the subset of sources that generate data in any given epoch) is independently carried a distance of d_1 hops to a common aggregation point; this data is then aggregated and carried another d_2 hops; finally it is delivered to the separate active sinks after an additional d_3 hops. Flooded interests and notifications are sent throughout the network (transmitted by all n nodes in the network), while directed interests and notifications and all reinforcements and data are sent through paths that lie on this tree.

3.4 Control and Application Data Sizes

There are essentially four kinds of messages sent within the network: interest notifications, event notifications, response/reinforcements, and application data. For tractability of analysis we don't model individual control and data packets, rather we model the traffic sent in each epoch as an aggregate. We assume that interest notifications sent by each sink in the pull-mechanism amount to S_I bytes per second over the epoch. In push and two-phase pull, exploratory messages amount to $S_E = \alpha S_D + S_C$ bytes per epoch, where α is the fraction of the total application data S_D for the epoch that is sent in the exploratory message and S_C is control overhead. S_R represents the rate of reinforcement messages sent in response to notifications.

All of these sizes are application-dependent, but for typical applications, interest messages are 60–120B, reinforcement messages are 100–160B. Data sizes are very application-dependent, but range from 60–500B/message for the the applications we employ. Control overhead in exploratory data is often small or zero, since the flags needed are already part of the header. Interests are sent every 30s and exploratory data every 90s. If for simplicity we assume that each interest, reinforcement, and data message is 120B in size and S_C is 4B and that sensors in the application generate data every 2s, this translates into the values shown in Table 1.

3.5 Metrics and Parameters

We will analyze the expected total control (setup) traffic C per epoch and the expected total application data traffic per epoch U , to compute the relative control overhead $O = C \cdot (U + C)^{-1}$. These overhead and traffic metrics will be evaluated as functions of several parameters: (i) the basic routing mechanism (push, two-phase pull, one-phase pull, or optimal rendezvous), (ii) the scenario (NAF, NAD, AF, AD) (iii) topological parameters (d_1, d_2, d_3, n, I, J), (iv) traffic parameters (p_j, l_i), as well as the data size parameters ($S_I, S_R, S_E, S_D, \alpha$).

4 Analysis

We now consider each scenario in turn and derive expressions for the traffic and overhead costs.

4.1 No Aggregation—Flooding (NAF)

We first consider the case when no aggregation is employed and interests/notifications are flooded throughout the network (i.e. no directed/geographic routing scheme is available). The total useful data in this case is

$$U^{NAF} = \sum_i l_i \sum_j p_j S_D (d_1 + d_2 + d_3) \quad (1)$$

Intuitively, $S_D(d_1 + d_2 + d_3)$ represents the cost of sending data over full distance, while the double summation captures which sources and sinks are interested.

a) Pull-based: In two-phase pull, the procedure is as follows: i) the sink floods interest, ii) the sources flood exploratory data in response iii) the sinks reinforce a specific path for each source's data and iv) the data is sent by the sources. In one-phase pull, we have i) the sink floods interest and ii) the sources respond with data by using knowledge of the reverse path of the flood. We must account for the fact that exploratory packets from the sources include useful data.

$$C_{2pull}^{NAF} = \sum_j p_j (S_I n + \sum_i l_i (S_E n + S_R (d_1 + d_2 + d_3))) - \alpha U^{NAF} \quad (2)$$

Here the first summation represents the cost of flooding interests, the S_E term in the second summation represents a flood of exploratory data, and the S_R term is reinforcements. Again, the final α term represents useful data piggybacked on control messages. The control overhead is therefore $C_{2pull}^{NAF} / (U^{NAF} + C_{2pull}^{NAF})$.

For one-phase pull, we have that the control traffic is simply the flooding of interests:

$$C_{1pull}^{NAF} = \sum_j p_j S_I n \quad (3)$$

b) Push based: The setup cost has to do with network-wide flooding of event notifications by all sources with data, and the direct (point-to-point) response of the corresponding interested sinks to these events. The pushed event includes exploratory data, so this must be accounted for in calculation of control traffic. Let C_{push}^{NAF} be the control, non-useful traffic in the case of NAF for the push paradigm. We have that

$$C_{push}^{NAF} = \sum_i l_i (S_E n + \sum_j p_j (d_1 + d_2 + d_3) S_R) - \alpha U^{NAF} \quad (4)$$

In this equation, the first summation represents the cost of flooding exploratory messages, the second the cost of reinforcements, and the last term represents the fact that some useful data is piggybacked on control messages (since exploratory data is both control and data). The relative control overhead is therefore given as $C_{push}^{NAF} / (U^{NAF} + C_{push}^{NAF})$. Note that the hybrid rendezvous-based scheme does not apply in this context if flooding is to be used, since by definition the rendezvous points are known a priori to both sources and sink.

Comparison of Push and One-phase Pull: We can now quantify the intuition that push is better than one-phase pull when there are fewer active sources and pull is better when there are fewer active sinks. We can derive an expression for the condition when the two are equivalent. let $l_i = l$ for all sources and $p_j = p$ for all sinks, then the average number of active sources is $\sum_i l_i = lI$ and the average number of active sinks is $\sum_j p_j = pJ$. For the NAF case, by equating expressions (3) and (4), assuming $\alpha = 0$ and $S_C = S_I$, we have that:

$$lI = \frac{pJ}{1 + pJ \frac{S_R d}{S_I n}} \quad (5)$$

Numerical Results: To illustrate these analytical results, we generated some plots based on numerical calculations. In these numerical calculations, the various parameters take on the following values: $I = J = 10$, $n = 100$, and the various traffic sizes per epoch are chosen as shown in table 1. The sink interest probability $p_j = p$, $\forall j$ is varied from 0 to 1, as is the source generation probability $l_i = l$, $\forall i$. (Unless otherwise noted, these parameters are used for all numerical results presented in this paper). The absolute setup costs C and relative overhead O for both one-phase pull and push-diffusion for the NAF case (from equations (3), and (4)) are plotted in figure 4.

Several inferences may be drawn from these figures. First, as can also be determined from equations (3), and (4), the setup costs for the one-phase pull mechanism and for the push mechanism increase linearly with respect to the sink interest rate p and the source event rate

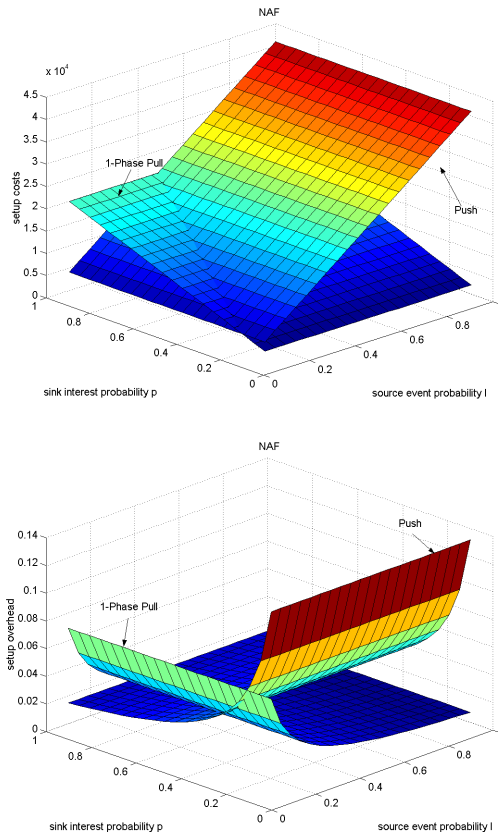


Figure 4: Setup traffic (left) and relative control overhead (right) in NAF case: push versus one-phase pull diffusion

l respectively. Second, one-phase pull outperforms the push mechanism when the source event rate is relatively high, while the reverse is true when the sink interest rate is relatively high. It is interesting to note that one-phase pull diffusion starts to outperform push-diffusion even when the sink interest rate is lower than the source event rate—this trade-off occurs because push-diffusion has additional overhead due to the reinforcement packets. Finally, these figures illustrate that it can be disastrous in terms of control traffic if the wrong version of diffusion routing is used for the application requirements. For example, when the source rate is very high and sink interest rate is very low, using push diffusion instead of pull diffusion can mean more than 80% difference in relative overhead in the scenario we have examined numerically. In the worst case topology, this difference in performance can be arbitrarily high.

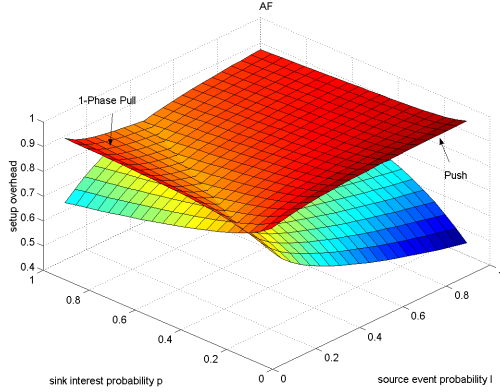


Figure 5: Relative control overhead in AF case: push versus one-phase pull diffusion

4.2 With Data Aggregation and Flooding (AF)

We assume that we can aggregate all data traffic from the sources into a single packet at the aggregation point at distance d_1 from the sources. Let P_{ij} be the probability that there are i “active” sources and j sinks in a given epoch. Then the useful data from all i sources is first carried separately for a distance d_1 , then aggregated and carried jointly for a distance d_2 , and finally delivered separately to each of the j interested sinks which are all an additional distance d_3 away in our model. Therefore,

$$U^{AF} = S_D \sum_{i=1}^I \sum_{j=1}^J P_{ij}(id_1 + d_2 + jd_3) \quad (6)$$

Now, since the setup/control traffic is not aggregated, the setup costs for the two-phase and one-phase pull and push algorithms in the AF case are identical to those in the NAF case. Thus expressions (2), (3), and (4) also apply to C_{2pull}^{AF} , C_{1pull}^{AF} , and C_{push}^{AF} respectively. Note that in the AF case (as with the NAF scenario), without geographic information to direct information to a rendezvous point, the hybrid rendezvous scheme cannot be implemented.

Comparison of Push and One-phase Pull: Even though the relative overheads are different with aggregation, the quantitative condition when the push and pull diffusion are equivalent is the same for the AF case as it is for the NAF scenario, i.e. equation (5) still holds.

Numerical Results: Figure 5 shows the fractional setup overhead for both push and one-phase pull diffusion for the AF scenario. As noted above, the absolute setup costs for the AF scenario are identical to that for the NAF scenario. However, the aggregation of data

reduces the number of data packets sent within the network, while making no impact on the setup costs. As a result the fractional setup overhead for both mechanisms is quite high (nearly 1 for most of the parameters studied in figure 5). This suggests that when data aggregation is employed, the relative rate at which interests and event notifications are flooded should be significantly reduced in order to minimize control overhead. In our model this would translate to increasing the value of S_D while keeping S_C , S_R and S_I the same.

4.3 No Aggregation - Directed Interests/Notifications (NAD)

If an underlying unicast scheme or geographic information allows for interests and notifications to be routed directly to the set of possible sources and possible sinks respectively, without the need for flooding, then the following are the pertinent expressions. Since data both with and without directed control traffic flows only on reinforced paths, it is not surprising that U^{NAD} is equal to U^{NAF} (Equation 1):

$$U^{NAD} = \sum_i l_i \sum_j p_j (d_1 + d_2 + d_3) S_D \quad (7)$$

a) Pull-based: the sinks direct their interests to all sources. In this context, two-phase pull does not make sense as once the interests are received, the data can be directly sent (using the available geographic information) by relevant sources to the pertinent sinks without need for an intermediate exploratory flooding-reinforcement phase.

$$C_{1pull}^{NAD} = \sum_j p_j S_I (d_1 + d_2 + d_3) I \quad (8)$$

The main difference between this and C_{1pull}^{NAF} (Equation 3) is the replacement of n (flooding) with $(d_1 + d_2 + d_3)I$.

b) Push-based: the event notification is sent directly to all possible sinks, and the responses from interested sinks is sent directly to notifying sources. Hence,

$$\begin{aligned} C_{push}^{NAD} &= \sum_i l_i (S_E (d_1 + d_2 + d_3) J \\ &+ \sum_j p_j ((d_1 + d_2 + d_3) S_R) \\ &- \alpha U^{NAD} \end{aligned} \quad (9)$$

c) Rendezvous: the sources send event notifications directly to the RP, the sinks direct interests to the RP, and the RP sends setup messages to all pertinent sources. Assume that the RP is located at the point that is closes

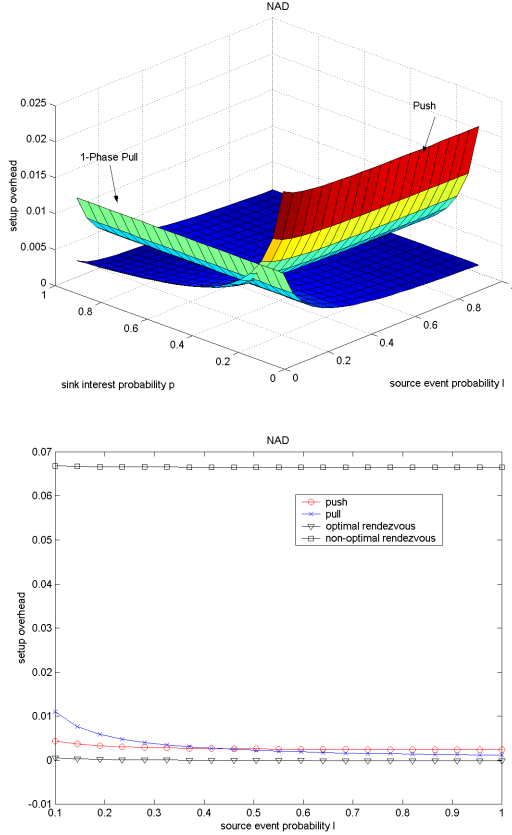


Figure 6: Relative control overhead in 3D (left) and for a 2D slice (right) in NAD case: push and one-phase pull diffusion, as well as optimal and non-optimal rendezvous

to all sources. Then,

$$C_{rendezvous}^{NAD} = \sum_i S_E l_i d_1 + \sum_j S_I p_j (d_3 + d_2) + \sum_j p_j \sum_i l_i d_1 S_I - \sum_i l_i \sum_j p_j (\alpha S_D) \theta$$

The first term indicates the cost of moving exploratory data to the RP where data converges (indicated “optimal RP” in Figure 3). The second term moves interests to the RP from the sinks. The third term carries interests to the active sources. Finally, the last term deducts the actual data in the exploratory packets that is sent towards the sinks.

Comparison of Push and One-phase Pull: Once again, we have that push is better than one-phase pull when there are fewer active sources and one-phase pull is better when there are fewer active sinks. Consider the point when the two are equivalent. As before, we let $l_i = l$ for all sources and $p_j = p$ for all sinks, and let $\alpha = 0$, $S_C = S_I$. For the NAD case, for both costs to

be equal it can be shown that the following must hold:

$$l = \frac{p}{1 + p \frac{S_R}{S_I}} \quad (11)$$

Numerical Results: Figure 6 shows how one-phase pull and push diffusion perform in terms of the relative overhead $O = \frac{C}{C+U}$ for different numbers of active sources and sinks. Figure 6 (right) compares pull, push and an optimal as well as a non-optimal rendezvous scheme. The principal observations are as follows.

As may be expected, the use of directed interests and event notifications significantly reduces the overhead of both one-phase pull and push mechanisms compared to flooding. Figure 6 also shows the performance of an optimal hybrid rendezvous scheme in which the rendezvous point is located on the shortest path between sources and sinks (as shown in figure 3) to minimize overhead. This suggests that optimal rendezvous techniques can combine the best aspects of pull and push to provide superior performance. However, in a practical rendezvous scheme, the rendezvous point may not be on the shortest path between sources and sinks and hence data packets are routed through more hops, leading to additional overhead. The curve for the non-optimal rendezvous scheme in figure 6 shows that significant additional overhead is incurred even when the route between source and sink through the RP is only one more hop longer than with the optimal rendezvous scheme. The poor performance of this scheme suggests in practice hybrid rendezvous schemes may not be efficient because of their sensitivity to the optimal RP placement.

4.4 With Data Aggregation and Directed Interests/Notifications(AD)

The expected useful data traffic per epoch is identical to the AF scenario since they both have data aggregation. Again, let P_{ij} be the probability that there are i “active” sources and j sinks in a given epoch. Then, just as in equation (6),

$$U^{AD} = S_D \sum_{i=1}^I \sum_{j=1}^J P_{ij} (i d_1 + d_2 + j d_3) \quad (12)$$

We consider the aggregation of data only, not of interest and exploratory packets. Therefore the setup costs for the AD case are identical to the setup costs of the NAD scenario described in section 4.3. Equations (8), (9), and (10) describe the setup costs for C_{1pull}^{AD} , C_{push}^{AD} and $C_{rendezvous}^{AD}$ respectively as well.

Comparison of Push and Pull: Again, we find that although the relative overheads are different, the push-pull equivalence condition for the AD scenario is the as for the NAD scenario shown in equation (11).

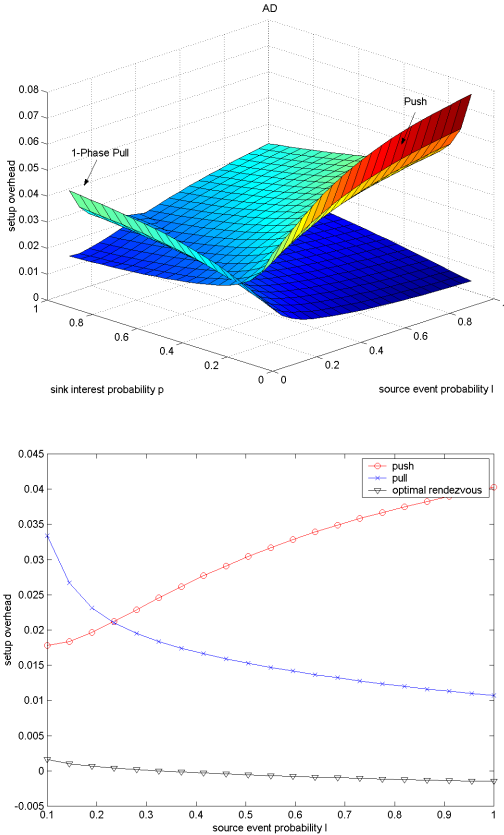


Figure 7: Relative control overhead in 3D (left) and for a 2D slice (right) in AD case: push, one-phase pull diffusion and optimal rendezvous

Numerical Results: Figure 7 shows numerically the performance of one-phase pull, push diffusion and an optimal rendezvous scheme for the AD scenario. Although not shown in this figure, we should note that for the AD scenario as well, the rendezvous scheme is found to have much worse performance if the RP is not optimally placed.

5 Related Work

Prior work analyzing diffusion is most closely related to this work. We have previously examined the impact of data aggregation on diffusion with both analysis and simulation [12]. We have also examined this question over regular topologies, with comparisons between diffusion, omniscient multicast, and flooding [10]. The conclusion of both analyses was that aggregation is central to efficient operation of data-centric routing. Work on geographic hash tables has included simple analysis of the communications costs of diffusion as compared to rendezvous approaches (identified there as data-centric

storage) [16]. In contrast to this prior work, our paper considers a wider range of diffusion algorithms and examines protocols both with and without aggregation.

Related to our work are mobile ad-hoc routing protocols, in which a central question is the choice between on-demand and a priori route computation [15]. However, these address-centric protocols are not directly concerned with the problem of matching sources and sinks; they defer that problem to resource discovery at the application layer.

Multicast protocols in wired networks have similar trade-offs to data dissemination protocols that match sources to sinks in sensor networks. Deering and Cheriton characterized multicast routing protocols with costs proportional to the numbers of sources and sinks [5], and later protocols such as PIM-SM examined rendezvous approaches [4]. Although Deering and Cheriton identified different protocols that were source- or sink-active, to our knowledge these differences have not been analyzed, probably because in many Internet multicast protocols, all nodes are both sources and sinks. Diffusion differs from this work in that it can exploit geographic constraints and aggregation, both of which we examine in this paper.

6 Conclusion

Different variants of the directed diffusion routing protocol—pull-based, push-based and hybrid rendezvous-based—have been developed to meet the diverse range of sensor networks applications. Moreover, some applications may exploit data-aggregation or geographic information. We analyzed these alternatives systematically using mathematical modelling to determine how well they match different application scenarios with different numbers of nodes, sources, sinks, data settings etc. We quantified the conditions under which push diffusion outperforms pull diffusion (and *vice versa*).

The results of this analysis also provide a number of useful design insights. We saw that the mismatch of routing algorithms to application scenario can result in drastically poor performance. For example, when the source rate is very high and sink interest rate is very low, using push diffusion instead of pull diffusion can mean more than 80% difference in relative overhead in the NAF scenario we studied numerically. In the worst-case topology, this difference in performance can be arbitrarily high.

We found that the relative control overhead for both push and pull algorithms was very dominant (close to 1) in the AF scenario. This is because the data is being aggregated while setup messages are being flooded. In

such scenarios, it is desirable to reduce interest and exploratory message. We also examined rendezvous techniques that may be used when geographic information is available. While we showed that they can theoretically outperform both push and pull, their performance is highly sensitive to the optimal placement of the rendezvous point. Our analysis suggests that it may be difficult to implement an efficient rendezvous technique in practice.

Finally, another contribution of this work is that the abstract parameterized models for data traffic and topology that we have defined and developed in this study can be used for other systematic studies of routing in sensor networks. As an ongoing effort, we are in the process of combining and comparing this analytical study with results from experiments with a real sensor network test-bed.

References

- [1] Deepak Ganesan and Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly resilient, energy efficient multipath routing in wireless sensor networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (poster)*, Long Beach, California, USA, October 2001. ACM.
- [2] Philippe Bonnet, Johannes Gehrke, and Praveen Shadri. Querying the physical world. *IEEE Personal Communications Magazine*, 7(5):10–15, October 2000.
- [3] David Braginsky and Deborah Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the First ACM Workshop on Sensor Networks and Applications*, pages 22–31, Atlanta, GA, USA, October 2002. ACM.
- [4] Stephen Deering, Deborah L. Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. The PIM architecture for wide-area multicast routing. *ACM/IEEE Transactions on Networking*, 4(2):153–162, April 1996.
- [5] Stephen E. Deering. Multicast routing in internetworks and extended LANs. In *Proceedings of the ACM SIGCOMM Conference*, pages 55–64, Stanford, CA, August 1988. ACM.
- [6] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999. ACM.
- [7] John Heidemann, Fabio Silva, and Deborah Estrin. Matching data dissemination algorithms to application requirements. under submission, April 2003.
- [8] John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proceedings of the Symposium on Operating Systems Principles*, pages 146–159, Chateau Lake Louise, Banff, Alberta, Canada, October 2001. ACM.
- [9] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 56–67, Boston, MA, USA, August 2000. ACM.
- [10] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *ACM/IEEE Transactions on Networking*, 11(1):2–16, February 2002.
- [11] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 243–254, Boston, Mass., USA, August 2000. ACM.
- [12] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. The impact of data aggregation in wireless sensor networks. In *Proceedings of the IEEE International Workshop on Distributed Event-Based Systems (DEBS)*, pages 575–578, Vienna, Austria, July 2002. IEEE.
- [13] Juan Liu, Jie Liu, James Reich, Patrick Cheung, and Feng Zhao. Distributed group management for track initiation and maintenance in target localization applications. In *Proceedings of the IEEE International Workshop on Information Processing in Sensor Networks*, page to appear, Palo Alto, California, USA, April 2003. IEEE.
- [14] Samuel Madden, Michael J. Franklin, Joseph Hellerstein, and Wei Hong. TAG: Tiny AGgregate queries in ad-hoc sensor networks. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, Boston, Massachusetts, USA, December 2002. USENIX.
- [15] Charles Perkins. *Ad Hoc Networking*. Addison-Wesley Pub Co, 2000.
- [16] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: A geographic hash table for data-centric storage. In *Proceedings of the ACM Workshop on Sensor Networks and Applications*, pages 78–87, Atlanta, Georgia, USA, September 2002. ACM.
- [17] Kung Yao, Ralph E. Hudson, Chris W. Reed, Daching Chen, and Flavio Lorenzelli. Blind beamforming on a randomly distributed sensor array system. *IEEE Journal of Selected Areas in Communication*, 16(8):1555–1567, October 1998.
- [18] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report TR-01-0023, University of California, Los Angeles, Computer Science Department, 2001.