

# Survival of the Fittest: Data Dissemination with Selective Negotiation in Wireless Sensor Networks

Xiaolong Zheng\*, Jiliang Wang<sup>†</sup>, Wei Dong<sup>‡</sup>, Yuan He<sup>†</sup>, Yunhao Liu<sup>†</sup>

\*Department of Computer Science and Engineering, HKUST

<sup>†</sup>MOE Key Lab for Information System Security, School of Software, TNLIST, Tsinghua University

<sup>‡</sup>College of Computer Science, Zhejiang University

{xiaolong, jiliang, he, yunhao}@greenorbs.com, dongw@zju.edu.cn

**Abstract**—Data dissemination is a building block of wireless sensor networks (WSNs). In order to guarantee the reliability, many existing works rely on a negotiation scheme, making senders and receivers negotiate the schedule of transmissions through a three-way handshake procedure. According to our observation, however, negotiation incurs long dissemination time and seriously defers the network wide convergence. On the other hand, the flooding approach, which is conventionally considered to be inefficient and energy-consuming, may facilitate data dissemination if appropriately designed. This motivates us to pursue a delicate tradeoff between negotiation and flooding in the data dissemination process. In this paper, we propose *SurF* (Survival of the Fittest), a data dissemination protocol which selectively adopts negotiation and leverages flooding opportunistically. How to capture and utilize the opportunities when negotiation should be used is a challenging issue. *SurF* incorporates a time-reliability model to estimate the time efficiencies of the two schemes (flooding vs. negotiation) and dynamically selects the fittest one to facilitate the dissemination process. We implement *SurF* based on TinyOS 2.1.1 and evaluate its performance with 40 TelosB nodes. The results show that *SurF*, while retaining the dissemination reliability, reduces the dissemination time by 40% in average, compared with the state-of-the-art protocols.

## I. INTRODUCTION

Wireless sensor networks (WSNs) [1] have been applied in a variety of application areas such as environment monitoring [2] [3], structural protection [4], military surveillance etc. Most WSNs, once deployed, are intended to operate unattended for a long period. During the lifetime of a WSN, it is often necessary to fix bugs, reconfigure system parameters, and upgrade the software in order to achieve satisfactory performance [5]. Data dissemination is a building block of WSNs to enable the above-mentioned important tasks.

Generally, data dissemination in WSNs must meet two requirements. First, it should be *reliable* despite unreliable wireless links in the network. Second, they should be *efficient* with respect to the dissemination time for the entire network reaches convergence. Note that long dissemination time usually means relatively lasting interrupts of the normal network operations and high energy consumption. It is thus significant to shorten the dissemination process.

Data dissemination attracts wide attentions in the WSN

community. A number of protocols have been proposed in recent years. As a representative example, Deluge [6] adopts a negotiation scheme proposed in [7] to guarantee the reliability and reduce redundant transmissions. Every Deluge node periodically broadcasts ADV messages to announce its own data of latest version. Neighboring nodes hear the ADV messages and send REQ messages to the ADV sender if a newer version is found. After receiving the REQ messages, the node starts sending DATA messages.

We notice that the negotiation scheme, although effective for ensuring the reliability of data delivery, incurs a large overhead in terms of dissemination time. In a typical experiment with two TelosB nodes transmitting 10KB data using Deluge, the time spent on negotiation contributes to 71% of the total dissemination time, which is far beyond our usual expectation.

This motivates us to *selectively* use the negotiation scheme *only when necessary* in the entire dissemination process, so as to improve the dissemination efficiency while retaining reliability. On the other hand, dissemination without negotiation (so-called *flooding*) makes each node probabilistically broadcasts a packet for  $n$  times.

We observe that (1) for certain success ratio of dissemination, flooding often has a much shorter dissemination time. For example, to disseminate 10K data, flooding takes less than 10 seconds while the negotiation-based protocol takes near 20 seconds in achieving 20% success ratio of dissemination. This is because flooding has advantages in the initial phase, when most of the nodes do not have the latest data. (2) On the other hand, for a higher success ratio of dissemination, flooding becomes inefficient because blind flooding without feedbacks tends to cause a large amount of redundancy but almost results in vain. In contrast, the use of negotiation in that phase may effectively avoid redundancy by explicitly requesting for the missing packets.

In this paper, we propose **SurF** (Survival of the Fittest), a data dissemination protocol which *selectively* utilizes negotiation to improve the efficiency. Flooding is adopted to substitute for negotiation *opportunistically*. *SurF* adaptively decides the best strategy and switches between flooding and negotiation to achieve improved dissemination efficiency

while remaining reliability.

A key issue in SurF’s design is to determine when and how nodes transit between the two schemes (flooding vs. negotiation). A bad transition point may result in longer dissemination time. SurF incorporates a time-reliability model to predict the time efficiencies of the two schemes. Based on that model, each SurF node estimates the potential benefit respectively brought by either of the two schemes and dynamically makes the decision on the most appropriate dissemination scheme in a distributed manner.

We implement SurF based on TinyOS 2.1.1 and evaluate its performance on a 40-nodes testbed.<sup>1</sup> The evaluation results demonstrate that (1) the model within SurF can accurately predict the completion time of two schemes. (2) SurF reduces the dissemination time by 40%, compared to Deluge.

The contributions of this paper are summarized as follows.

- We find that the selective use of negotiation and opportunistic leveraging of flooding will improve the dissemination time without harming reliability.
- We adopt an accurate time-reliability model to estimate and predict the performance of different schemes to capture the opportunities of selective negotiation.
- We implement SurF and evaluate its performance through experiments on real testbeds. The results demonstrate the advantages of SurF in terms of dissemination time, compared with Deluge.

The rest of this paper is organized as follows. Section II discusses the motivation behind this work. Section III describes the analytical model for estimating the dissemination performance. Section IV elaborates on the design of SurF. Section V presents the evaluation results. Section VI discusses the related work, and Section VII concludes this paper.

## II. MOTIVATION

In this section, we present the experimental observations that motivates our work. We use testbed experiments consisting of 40 TelosB nodes to study the performance of flooding and negotiation-based dissemination.

- Flooding. Each node performs probabilistic flooding with probability  $p$  and number of flooding  $n$  [8], i.e., upon receiving the first new packet, a node broadcasts the packet with probability  $p$  for  $n$  times. For the sink node,  $p = 1$ , and for other nodes  $p = 0.9$ . We tune  $n$  to achieve certain level of success ratio of dissemination.
- Negotiation-based dissemination. We use the default Deluge protocol with a page size of 48 packets.

We disseminate 5 pages in the network. We define the *node reliability* at time  $t$  as the success ratio of dissemination on this node, i.e., the ratio of the number of unique received

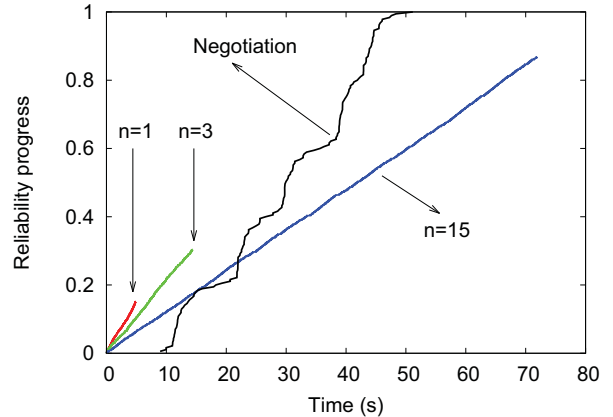


Figure 1. Reliability progress

packets before time  $t$  to the total number of needed packets (i.e.  $48 \times 5$ ). We define the *network reliability* as the average of all node reliability. When discussing the reliability progress in the remaining of this paper, it refers to the success ratio of dissemination.

Figure 1 presents the time-reliability curve obtained from our experiments. We see that (1) For a small number  $n$ , flooding cannot achieve a high reliability since blind flooding with no feedback repeats infinite times for a reliable dissemination in theory. Hence, negotiation is indispensable to guarantee a high reliability. (2) However, for a certain level reliability, e.g.,  $< 30\%$ , flooding (with  $n = 1, 3$ ) has a much shorter completion time compared to the negotiation-based dissemination. (3) For a high level of reliability, e.g.,  $> 80\%$ , flooding (with  $n = 15$ ) has a longer completion time than negotiation-based dissemination. (4) A naive combination of two schemes with a fixed  $n$  is possible to shorten the completion time of dissemination. However, it cannot work well when considering the dynamic networks. We will show these statements in Section III from analysis and in Section V from experiments.

## III. ANALYTICAL MODELS

In this section, we present the analytical model for SurF, which seizes the opportune moment when negotiation should be put to use to minimize the completion time. We analyze the completion time of flooding-based methods, negotiation-based methods and also the combination of these two schemes. By analyzing the completion time, we show that a combination of flooding and negotiation can shorten the completion time. And we further show that an optimal decision about the transition point between two schemes can lead to minimized completion time in single hop.

To allow distributed computation at each node, we study the time-reliability model of SurF in a neighborhood. In Section IV-F, we also analyze the multi-hop performance improvement. And we further show SurF’s local optimality

<sup>1</sup>The code is publicly available at [http://www.cse.ust.hk/~xzhenaga/SurF\\_DataDissemination.rar](http://www.cse.ust.hk/~xzhenaga/SurF_DataDissemination.rar)

often leads to considerable network performance improvement through experiments, presented in Section V. Here we just simply regard the local optimality as the objective of optimization. Therefore, we estimate  $T_{ij}^S(n, \phi, q_{ij})$ , the completion time at node  $i$  with SurF, given the number of flooding  $n$ , the reliability requirement  $\phi$ , and the worst link quality  $q_{ij}$  from  $i$  to one of its neighbors  $j$ . SurF minimizes the completion time by finding the optimal transition point between two schemes, based on the analytical model.

Some notations used in our design are listed below.

- $N$ , the number of packets in one page.
- $q_{ij}$ , the link quality between node  $i$  and its neighbor  $j$ .
- $p_i$ , the rebroadcasting probability of node  $i$  in flooding.
- $T_{pkt}$ , the transmission time per packet.
- $T_{back}$ , the expected back-off time before sending out a packet.
- $R_j^0$ , the initial node reliability at  $j$ .
- $N_{supp}$ , the expected number of suppressed ADVs due to the suppression scheme used in negotiation.
- $\tau_l$ , the expected time between two successive ADVs.
- $\tau_r$ , the expected time between two successive REQs.
- $R_j(n, R_j^0)$  denotes the reliability that node  $j$  can obtain after the flooding/negotiation of node  $i$  with  $n$  times, given that  $j$  already has a reliability of  $R_j^0$ . It is:

$$R_j(n, R_j^0) = 1 - (1 - R_j^0)(1 - p_i q_{ij})^n \quad (1)$$

For all nodes in negotiation and sink in flooding,  $p_i = 1$ . For other nodes in flooding,  $p_i \in (0, 1]$ .

- $n(\phi, R_j^0)$ , the number of transmission rounds. We define one transmission round as one transmission of a page together with the corresponding control process. Based on Eq. 1, it is:

$$n(\phi, R_j^0) = \left\lceil \frac{\log(1 - \phi) - \log(1 - R_j^0)}{\log(1 - p_i q_{ij})} \right\rceil \quad (2)$$

- $n_F(\phi, R_j^0)$ , the transmission rounds to achieve the required reliability  $\phi$  by only flooding, given the initial node reliability  $R_j^0$ .
- $T_{ij}^F(n)$ , the time needed by flooding from  $i$  to  $j$ , with  $n$  rounds.

$$T_{ij}^F(n) = nN(T_{pkt} + T_{back}) \quad (3)$$

- $T_{ij}^N(\phi, R_j, q_{ij})$ , the time needed by negotiation from  $i$  to  $j$ , given the reliability requirement of  $\phi$ , the already achieved reliability  $R_j$ , and the link quality  $q_{ij}$ .

From these notations,  $T_{ij}^S(n, \phi, q_{ij})$  can be estimated as:

$$T_{ij}^S(n, \phi, q_{ij}) = T_{ij}^F(n) + T_{ij}^N(\phi, R_j(n, 0), q_{ij}) \quad (4)$$

where  $T_{ij}^F(n)$  is given in Eq. 3 and  $T_{ij}^N(\phi, R_j(n, 0), q_{ij})$  is analyzed as follows. In the following analysis, we omit the parameters for simplicity when no ambiguity occurred.

As shown in Figure 2, the time of negotiation scheme  $T_{ij}^N$  is composed by three parts: (1)  $T_{ij}^{ADV}(q_{ij})$ , the time

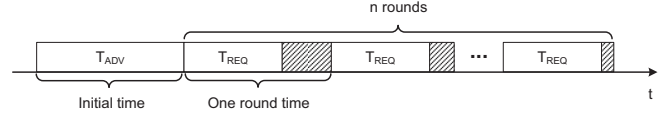


Figure 2. Composition of completion time of negotiation

of the initial ADV from a sender; (2)  $n_2 T_{ij}^{REQ}(q_{ij})$ , the time of  $n_2$  rounds REQ transmission, taking  $T_{ij}^{REQ}(q_{ij})$  time for each round; (3)  $T_{ij}^{DATA}(R_j(n, 0), q_{ij})$ , the time of DATA transmission. That is:

$$T_{ij}^N(\phi, R_j(n, 0), q_{ij}) = T_{ij}^{ADV} + n_2 \cdot T_{ij}^{REQ} + T_{ij}^{DATA} \quad (5)$$

where  $n_2$  is the transmission rounds of negotiation to meet the reliability of  $\phi$ , given the reliability already achieved by flooding with  $n$  rounds,  $R_j(n, 0)$ . That is:

$$n_2(\phi, R_j(n, 0)) = \left\lceil \frac{\log(1 - \phi) - \log(1 - R_j(n, 0))}{\log(1 - p_i q_{ij})} \right\rceil \quad (6)$$

The time of initial ADV from node  $i$  to  $j$  is the expected time that  $j$  receives an ADV from  $i$ , which is given by

$$T_{ij}^{ADV}(q_{ij}) = \tau_l \cdot \left( \frac{1}{q_{ij}} - 1 + N_{supp} \right) \quad (7)$$

The time of REQ per round is the expected time node  $i$  receives a REQ after  $j$  receives an ADV:

$$T_{ij}^{REQ}(q_{ij}) = \tau_r \cdot \left( \frac{1}{q_{ij}} - 1 \right) + E[N_{ADV}] \cdot T_{ij}^{ADV}(q_{ij}) \quad (8)$$

where  $E[N_{ADV}]$  is the expected number of additional ADVs needed in one round. Note that nodes cannot send REQ unlimitedly in one round, it stops trying after  $\lambda$  times, the maximum number one can try before next ADV is heard. Suppose  $X$  is a random variable which represents the number of REQs transmitted for one page. Thus,  $X \sim G(p_{ij})$ . Then the expected number of additional ADVs during one page's dissemination is:

$$\begin{aligned} E[N_{ADV}] &= \sum_{k=0}^{\infty} k \cdot P(N_{ADV} = k) \\ &= \sum_{k=0}^{\infty} k \cdot P(k\lambda < X \leq (k+1)\lambda) \\ &= \sum_{k=0}^{\infty} \left( k \cdot \sum_{m=k\lambda+1}^{k\lambda+\lambda} P(X = m) \right) \\ &= \frac{(1 - q_{ij})^\lambda}{(1 - (1 - q_{ij})^\lambda)} \end{aligned} \quad (9)$$

Even though the DATA transmission scatters in different rounds, as the dashed areas shown in Figure 2, the time can be measured by the expected number of transmitted packets for this page. Given the reliability that has already been achieved by flooding with  $n$  times,  $R_j(n, 0)$ , and the link

quality between  $i$  and  $j$ , the time of DATA is the expected time of transmission for this page, which can be written as

$$T_{ij}^{DATA}(R_j(n,0),q_{ij}) = \frac{(1-R_j(n,0))}{q_{ij}}N(T_{back} + T_{pkt}) \quad (10)$$

Note that Eq. 5 gives the completion time of SurF in the case where  $n_2 \neq 0$ . We can re-express the completion time in a more general form. That is:

$$T_{ij}^S(n,\phi,q_{ij}) = \begin{cases} T_{ij}^N(\phi,0,q_{ij}), & n=0 \\ T_{ij}^F(n) + T_{ij}^N(\phi,R_j(n,0),q_{ij}), & 0 < n < n_F \\ T_{ij}^F(n_F), & n=n_F \end{cases}$$

When  $n=0$ , the integration retrogrades in the negotiation-based methods. When  $0 < n < n_F$ , the integration leverages two schemes. When  $n=n_F$ , the integration turns into flooding. However, if the required reliability is 100%, then  $n_F \rightarrow \infty$ , i.e., the negotiation is a must to guarantee reliability. Therefore,  $n \in [0, n_F]$ .

Note that when SurF decides flooding is not efficient, it leverages negotiation to substitute for it. Hence, to decide the optimal transition point for minimal completion time is equivalent to decide the optimal number of flooding  $n$ . From the models and analysis, the benefit SurF can bring is presented and the demand of optimization of flooding rounds  $n$  is revealed. However, how to exploit the benefits and find the optimal  $n$  in a distributed manner need to be designed.

#### IV. DESIGN

Based on the analytical results, we present the design of SurF. SurF adopts several design principles: (1) selecting the optimal strategy to avoid unnecessary negotiation for minimum completion time; (2) estimating parameters and making decision in a distributed manner to adapt to the dynamic networks; (3) leveraging segmentation and pipelining to exploit the spatial multiplexing for scalability.

##### A. Overview

Figure 3 shows the state transition diagram of SurF. At the beginning, each node stays at *maintain* state. Nodes in *maintain* state send out the periodical ADV messages to keep the network consistent. When a node receives an ADV with later version number, it immediately broadcasts the ADV with its own version number. If a node receive an ADV with outdated version number, SurF decides this node gets a updated page and it is responsible for disseminating this page. To leverage the spatial multiplexing, SurF also exploits segmentation and pipelining. The code image is divided into fixed-size pages. Each page is disseminated sequentially by SurF's algorithm, as following.

When a node gets an updated page, it estimates the benefit of different strategies based on our analysis model. According to the estimation result, the node will switch to the *negotiation* state or *flooding* state. If a node switches

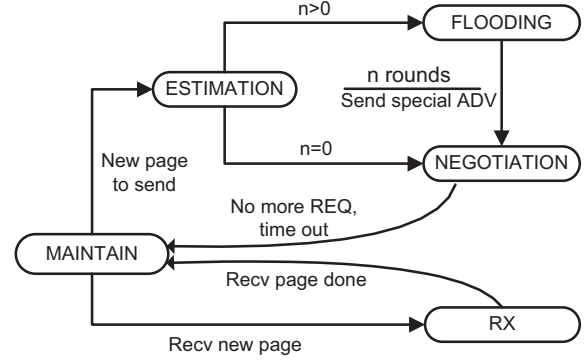


Figure 3. State transition diagram of SurF

to the *flooding* state, the node firstly floods the page for  $n$  rounds, where  $n$  is the estimated optimal rounds. Afterwards, the node turns into the *negotiation* state to finish the dissemination by negotiation scheme to guarantee the reliability. If a node in *negotiation* does not receive any REQ message for a certain time period, it returns to *maintain* state. When a SurF node receives packets of new page, it transits into *rx* state. And it will not return *maintain* state until receiving the whole page.

Based on the main working flow, SurF has three key components: (1) parameters estimation component; (2) optimal strategy selection component; (3) state switching component. In the following subsections, we will present SurF's detailed designs.

##### B. Parameters estimation

There are two parameters needed to estimate in our design. One is the link qualities of neighbor nodes and the other one is the number of suppressed ADVs.

1) *Link qualities*: The link quality is a key parameter in our design to select the fittest strategy. SurF incorporates the LEEP link estimation protocol [9] to estimate the link qualities. The LEEP header contains a sequence number to help the receiver estimate the inbound link quality from one neighbor by counting the successfully received packets among all the packets that neighbor transmits. Outbound link qualities can be obtained by the advertisements from its neighbors, which announce its inbound link qualities. We attach the LEEP header to all the messages in SurF, including flooding data packets, ADV, REQ, and DATA messages. The plenty of data packets effectively provide the good basis for estimating the inbound link qualities. SurF integrates the information of outbound link qualities into ADV messages instead of extra advertisement packets. Then neighbors can learn its outbound link qualities by neighbors' periodical ADV messages.

2) *Number of suppressed ADVs*: In the design of the negotiation scheme, the sender suppresses the ADV packet if similar ADV packets are overheard in its neighborhood. Due

to this scheme,  $T_{ADV}$  may be delayed since each suppression results in one additional ADV waiting time. In previous work such as [6], a linear topology is assumed and therefore  $N_{supp}$  is assumed to be 1. However, the linear topology usually does not meet the practical settings of real deployed systems. Therefore, the accurate estimation of  $N_{supp}$  in a general topology is crucial to estimate the time of negotiation scheme. SurF measures  $N_{supp}$  by measuring the expected number of upstream neighbors who have the chances to suppress the ADVs, taking link qualities into consideration. The expected number of suppressed ADVs of node  $i$  is measured as:

$$N_{supp} = \sum_{j \in M_i^{UP}} 1 \cdot q_{ji} \quad (11)$$

where  $M_i^{UP}$  is the set of upstream neighbors. Nodes learn the knowledge that whether a neighbor is upstream or downstream by overhearing. During the period that the node is in  $rx$  state, it firstly receives packets by flooding. Its upstream neighbors is expected to flood packets during the *flooding* phase. Hence, this node can overhear the transmissions of upstream nodes. Note that periodical ADV messages let nodes able to maintain the neighboring table. Thus a node can decide its downstream neighbors after the upstream neighbors are detected.

### C. Optimal strategy selection

The decentralized strategy selection component is to decide the fittest strategy to minimize the completion time. Based on *Property 1, 2 and 3* and the estimated parameters, we can calculate and select the best strategy, i.e., the optimal flooding rounds  $n$ . The overall operations on each node for optimal strategy selection are as follows.

*Step 1:* Get the link qualities and  $N_{supp}$  from the parameters estimation component;

*Step 2:* Calculate the optimal  $n$  based on *Property 1-3*;

*Step 3:* If  $n > 0$ , switch to *flooding*; If  $n = 0$ , switch to *negotiation*;

*Step 4:* If a new page is received, goto *Step 1*.

There are two problems to address while selecting the optimal strategy in SurF. First, the diversity of link qualities should be considered. Normally, the completion time of data dissemination is decided by the completion time of the last node. However, using the worst link quality to select strategy is not appropriate since it may incur too many redundancy flooding rounds. To address this problem, we use the median of the link qualities for all neighbors to decide the optimal  $n$ . By using this approach, the flooding approach can quickly process and not influenced by some extremely low link qualities. Note that SurF leverages LEEP to do the link estimation. Each time SurF makes decision about optimal  $n$ , the link qualities of neighbor nodes in neighboring table will be sorted. Then the median link quality is picked out for calculation.

Second, the diversity of node statuses should be taken into consideration. The downstream neighbors of a sender may be able to receive the data packets earlier than the sender from other paths. The sender should not consider covering those nodes. In our design, we obtain the neighbors' statuses by overhearing. When a downstream neighbor is sending the same page as the sender is sending/to send, the sender will omit this node from the neighbor set when calculating the optimal  $n$  for this page. Due to the status diversity on the receivers, if the newly estimated  $n$  is smaller than the flooding rounds already conducted by the sender, the sender switches to *negotiation* state.

### D. State switching

By optimal strategy selection, each node can select the optimal strategy during the dissemination process and switch between the two strategies. Switching between different strategies should be carefully designed to fulfill the following requirements.

First, the switching should be efficient. When nodes transit from *flooding* state to *negotiation* state, the sender periodically sends out ADV to set up the negotiation to receivers. However, this introduces significant overhead since the initial ADV negotiation time is quite long, especially when link qualities are poor or  $N_{supp}$  is large. To improve the efficiency, we subtly use the data packets in flooding to serve as the initial ADVs, significantly reducing the overhead of ADVs.

Second, the switching should be error-resilient. As soon as a node transits into *negotiation* state, it broadcasts multiple special ADVs to notify its neighbors about the transition. The receivers then send out REQs to request the missing packets. After the multiple special ADVs, the sender will periodically send out the ADV messages to maintain consistent. Hence, even if the receiver lost all the special ADVs, it still can require the missing packets by periodical ADVs.

By this way, SurF can significant reduce the negotiation time while still guaranteeing the delivery of ADVs to the receivers.

### E. Negotiation-based and flooding methods in SurF

SurF can be incorporated to any flooding and negotiation-based protocols. In current design, we use an improved version of Deluge, standard dissemination protocol of TinyOS. We improve it by reducing the initial ADV packets in Deluge by leveraging data packets in flooding.

We use the probabilistic flooding [8] as the dissemination scheme in flooding phase. Probabilistic flooding is a light-weight broadcast scheme which alleviates broadcast storm problem from two aspects: (1) mitigating collision by random back-off scheme; (2) reducing the redundancy by probabilistic rebroadcasting. The random back-off time used in SurF is 10-25 ms. The rebroadcasting probability is set to 0.9.



Figure 4. Testbed

### F. Multi-hop performance

In our design, each node selects the optimal strategy based on the local information. Nevertheless, we show that our approach can also lead to network-wide improvement. The completion time of the multi-hop network is determined by the last completed node. Suppose the path from the sink to the last completed node is:  $PT_{st} = (r_0, r_1, \dots, r_{|PT_{st}|})$ , where  $r_0$  is sink  $s$  and  $r_{|PT_{st}|}$  is the last node  $t$ . Considering interference between different nodes, when node  $t$  receives a page, the next page is at least three hops away with pipelining. Therefore, if the  $|PT_{st}| = h > 3$ , then the completion time in multi-hop  $T_H$  can be depicted as:

$$T_H = \sum_{i=0}^h T_{r_{i-1}r_i}^H + n_{pg} \cdot \sum_{i=h-2}^h T_{r_{i-1}r_i}^H \quad (12)$$

where  $n_{pg}$  is the total number of pages of the code image. By Eq. 12, we can see that the total completion time in multi-hop is decided by the completion time in single hop. Hence, the local optimization can also lead to global improvement on the completion time in the network. The evaluation results in Section V will further validate the effectiveness of SurF.

## V. EVALUATION

We implement SurF in TinyOS 2.1.1 with TelosB motes. In this section, we evaluate its performance. First, we validate the accuracy of the model proposed in Section III. Second, we compare SurF with Deluge on a real WSN testbed to show SurF's performance improvements. Finally, we discuss the comparison with other protocols and the energy consumption.

### A. Evaluation methodology

We conduct our evaluation in a testbed consisting of  $5 \times 8$  TelosB nodes, as shown in Figure 4. The sink is placed at the bottom left corner. We set the radio power level to 1 to emulate multi-hop transmissions. In the experiments, we

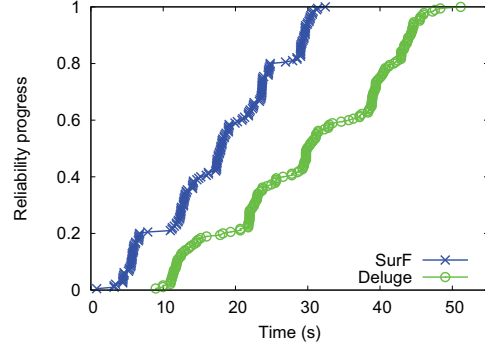


Figure 5. Completion time of SurF and Deluge

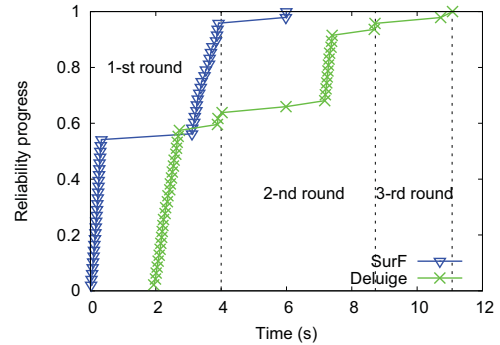


Figure 6. Reliability progress of one page

use Deluge with its default configurations (e.g., 48 packets per page).

We inject a data image of 5 pages (i.e., approximately 5KB) into the sink and disseminate the data from the sink. We record the dissemination progress of Deluge and SurF during the experiments. We plot the time-reliability curves to study the performance.

To get the performance metrics of different protocols, we write the statistic reporting component in our previous work [10]. It can report counters of different events as well as each individual event. Synchronization is performed at the start of each experiment. The sink will broadcast the time synchronization packets with the maximum power. The remaining nodes can thus synchronize to the sink after receiving the synchronization packets. After each experiment, we collect all the local logs via serial communications.

### B. Evaluation results

Figure 5 compares the performance of SurF and Deluge in our testbed. It takes 32.4s for SurF and 51.2s for Deluge to disseminate 5 pages. In this case, SurF achieves about 40% performance improvement, compared to Deluge. SurF shortens the completion time for the following two reasons. First, SurF has shorter inter-page negotiation time (e.g., time spent in the ADV phase) compared to Deluge. This is because SurF nodes flood the new page after the reception without

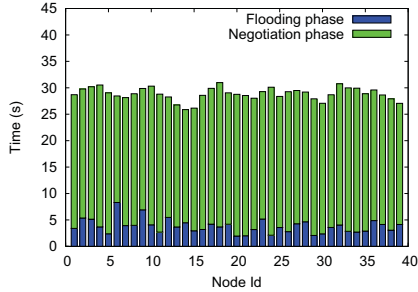


Figure 7. Time composition of flooding phase and negotiation phase on nodes

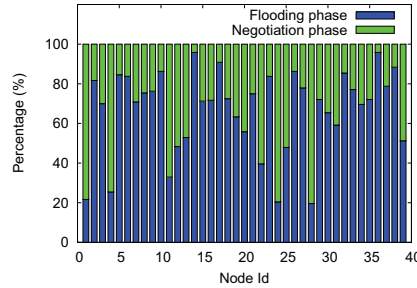


Figure 8. Percentage of packets received in the flooding phase and negotiation phase

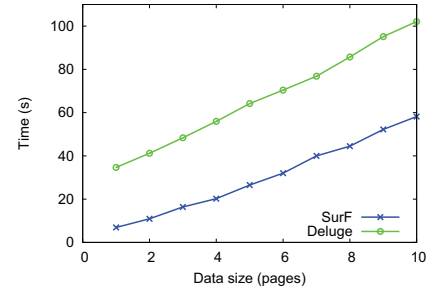


Figure 9. Completion time for varying data sizes

extra negotiation time to initiate actual data transmissions. Second, SurF has a shorter page dissemination time. Due to the flooding strategy it adopts, SurF can achieve a certain level of reliability quickly. Hence, the transmission rounds of negotiation can be greatly reduced, resulting in reduced dissemination time.

Actually, SurF shortens the completion chiefly by reducing the negotiation overhead. The idle-waiting time during negotiation is better utilized in SurF, by useful data transmissions. For example, Figure 6 depicts the reliability progress of the fourth page between node 0 and node 1 where node 0 is the sink. We can see that Deluge needs three rounds of negotiation to finish the dissemination. On the other hand, in SurF, the initial flooding achieves 54% reliability in 1s, leaving 22 missing packets to be covered in the negotiation phase. SurF reduces one negotiation rounds for the transmission of page 4. The first negotiation in Deluge is replaced by flooding in SurF since the negotiation in initial phase is inefficient.

Figure 7 depicts the fraction of the times of flooding phase and negotiation phase on each individual node with SurF. It shows the times of 39 nodes (excluding the sink, node 0). In the experiment, flooding one page needs 0.73s on average. Hence, we can see that all SurF nodes use flooding for at least one time. However, the strategies on different nodes are different due to various network conditions. Among all the nodes, node 6 (N6) has the longest flooding time of 8.3s. We inspect the neighbors of N6 to find out the underlying causes. It turns out that N6 has 3 upstream neighbors with good link qualities. Hence,  $N_{supp} = 3$ . The median link quality of its downstream neighbors is 61%. Based on the decision making algorithm, it decides flooding twice for each page.

Figure 8 depicts the number of packets received during flooding phase and negotiation phase. Combining Figure 7, we see that even though the time spent in flooding is short, the reliability it achieves is high. By leveraging flooding opportunistically, SurF selectively uses the negotiation to reduce its overhead while retaining the reliability.

We further conduct other experiments to study the scalability of SurF and Deluge. The experiment is repeated to

Table I  
COMPARISON OF SURF TO EXISTING PROTOCOLS

Protocols	# of nodes	Data size(KB)	Reduction factor
MNP( [11], 2005)	100	5.6	1.21
Rateless Deluge( [12], 2008)	20	0.7	1.47
ReXOR( [13], 2011)	16	4	1.53
ECD( [10], 2011)	25	10	1.44
<b>SurF</b>	40	10	1.75

present the average completion time. Figure 9 shows the completion times varied with different code image size. We can see that the completion times of both Deluge and SurF show the linearly increase. SurF achieves a much better performance than Deluge.

**Comparison with Existing Protocols.** We compare SurF to other existing works from the completion time. Completion time is the chief optimizing goal due to it decides the time application system works regularly again. Besides, since radio is the major energy consumption source, less radio on time means less energy consumption. In existing works, radio is usually kept on during dissemination. Hence, we could also leverage completion time as a measurement of energy consumption of dissemination.

We adopt the reduction factor to compare SurF with other existing protocols [14]. Deluge is the default dissemination protocol in TinyOS, which is widely applied in many real WSN systems. Thus we use Deluge as the baseline of comparison. Table I presents the reduction factor achieved by each protocol compared to Deluge. We can see that SurF apparently outperforms the state-of-the-arts protocols. Note that MNP has no report of completion time on testbeds. The items listed in table are obtained from simulations, which do not consider practical conditions, e.g collisions. Besides, it is already shown MNP is less efficient than ECD in [10]. Hence, despite that MNP used to be evaluated in a larger network, it is actually less efficient than SurF. Another fact worth noticing is that most existing protocols are compared with Deluge under TinyOS 1.x, which is actually slower than Deluge built in TinyOS 2.x. We used Deluge built in TinyOS 2.x for comparisons with SurF, which means

SurF outperforms the existing protocols much more than the reduction factor listed in Table I.

Moreover, the sleeping techniques can also be adopted in SurF to further reduce the energy consumption. Note that SurF only needs to keep radio on in the flooding-phase, which accounts for only a small fraction of the total completion time (as shown in Figure 7).

## VI. RELATED WORK

In data dissemination, the negotiation scheme is widely adopted to guarantee the reliability of data dissemination. Based on whether there is a dedicated structure construction for dissemination, the negotiation-based data dissemination protocols can be classified into two categories: the structure-less protocols and the structure-based protocols.

The representative of structure-less protocols is Deluge [6]. It adopts segmentation and pipelining technique for spatial multiplexing, and employs a three-way handshake negotiation scheme to guarantee the reliability of dissemination. MNP [11] designs a sender selection scheme to pick out the nodes who receive the most REQs as the next forwarder. ECD [10] is a recent work which improves the sender selection algorithm in MNP by taking link quality into consideration. It supports dynamic packet sizes to fit different PHY rate radios. SurF is similar to a structure-less protocol with regard to the ability to work well without structure construction involving. However, SurF differentiates with aforementioned works by avoiding unnecessary negotiation processes and selectively adopting negotiation to minimize the completion time of dissemination.

The other category is structure-based. CORD [15] is a representative of those protocols. CORD builds a connected dominating set as the backbone of the network, and employs a two-phase dissemination protocol. The nodes selected in the backbone network is called core nodes. In the first phase, the code is disseminated in along the backbone network and the non-core nodes passively listen to the transmissions. Then the non-core nodes ask their dominated core nodes for the missing packets in the second phase. Sprinkler [16] is another structure-based approach. It constructs a minimum connected dominating set based on the network topology for dissemination. SurF does not require such structures to function, avoiding any overhead to construct dissemination structures. Besides, the structure-less feature of SurF is more suitable for dynamic networks.

Flooding is the representative of non-negotiation methods. It is shown that blind flooding usually takes risk of broadcast storm [8]. In [8], the author proposes five schemes to relief the broadcast storm problem. In [17], the authors present the adaptive forms of the above mentioned five schemes. In Smart Gossip [18], an adaptive probabilistic flooding protocol is proposed. It automatically adjusts the rebroadcasting probability to adapt to the dynamic underlying network topology. DCB [19] adopts sender selection to avoid too

much redundant broadcasting and improves the delivery ratio. Being aware of the information of neighbors in two-hop, DCB can select senders to make every node has two chances to receive the packet. In [20], the parallelization of all possible interference-free relays in broadcasting is maximized to improve the pipeline process.

SurF makes use of the advantageous features of the above protocols (i.e., fast prorogation), but differs from them by achieving high reliability. The non-negotiation protocols cannot guarantee the reliability on account of the absence of ARQ scheme.

Coding is another technique that can be introduced into data dissemination. Rateless Deluge [12], SYNAPSE++ [21] and ReXOR [13] are all coding-based protocols. In those protocols, the sender encodes the message using certain coding techniques and transmits the encoded packets. After receiving sufficiently many encoded packets, a receiver recovers the data. Hence, receivers can send NACK which just declares the number of missing packets instead of the form of bit vectors with the specific information of missing packets. However, a limitation of coding-based protocols is the time consumption of decoding which may incur long completion time. Our work is orthogonal to those works and the coding technique can also be used in SurF to transmit the data packets.

Multichannel is another way to achieve spatial multiplexing. The authors in [22] propose the structure policies that achieve an asymptotically optimal average delay in the multichannel disseminations. The authors in [23] study the problem that how schedule data retrieval in multiple channels to maximize the number of downloads given a deadline. These works leverage multichannel to multiplex for improving dissemination efficiency. Our work is orthogonal to those works since the dissemination in each channel is still carried out by flooding or negotiation-based methods. SurF can also be integrated with the multichannel technique.

## VII. CONCLUSION

In this paper, we present SurF, a novel data dissemination protocol which selectively uses negotiation and opportunistically adopts flooding. We find that neither flooding nor negotiation is efficient when only one of them is used during the whole process. We then design SurF to effectively integrate these two schemes for shorter completion time. SurF selectively uses the negotiation scheme, i.e., only when necessary instead of through the entire dissemination process. Based on an accurate analytical model, SurF predicts the time efficiency of two schemes (flooding and negotiation) and adaptively selects the fittest strategy to disseminate data. SurF designs a distributed decision making algorithm to calculate the optimal transition points between two schemes for minimized completion time in single hop.

We implement SurF in TinyOS 2.1.1 with TelosB motes and evaluate its performance through experiments on real



testbeds. The experimental results shows the effectiveness of SurF. By reducing the negotiation overhead, SurF shortens the completion time while still retaining high reliability. Moreover, SurF does not depend on special protocols. It can be incorporated with other flooding-based and negotiation-based methods. In the future, we plan to integrate other protocols into SurF and further study the potential performance improvements.

#### ACKNOWLEDGEMENT

This work is supported in part by National Basic Research Program (973 program) under grant 2014CB347800 and 2011CB302705, NSFC under grants 61202402, 61170213 and 61202359, the Fundamental Research Funds for the Central Universities under grant 2012QNA5007, China Post doctoral Science Foundation under grant 2012M520013, and the Research Fund for the Doctoral Program of Higher Education of China under grant 20120101120179.

#### REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy closure estimates with greenorbs: Sustainable sensing in the forest," in *Proceedings of ACM SenSys*, 2009.
- [3] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. LI, and Y. Liu, "Citysee: Urban co2 monitoring with sensors," in *Proceedings of IEEE INFOCOM*, 2012.
- [4] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of ACM SenSys*, 2004.
- [5] L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys*, 2011.
- [6] J. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the ACM SenSys*, 2004.
- [7] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 169–185, 2002.
- [8] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM MobiCom*, 1999.
- [9] "Tinyos tep 124: The link estimation exchange protocol (leep)," <http://www.tinyos.net/tinyos-2.x/doc/html/tep124.html>.
- [10] W. Dong, Y. Liu, C. Wang, X. Liu, C. Chen, and J. Bu, "Link quality aware code dissemination in wireless sensor networks," in *Proceedings of IEEE ICNP*, 2011.
- [11] S. Kulkarni and L. Wang, "Mnp: Multihop network reprogramming service for sensor networks," in *Proceedings of IEEE ICDCS*, 2005.
- [12] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes," in *Proceedings of ACM/IEEE IPSN*, 2008.
- [13] W. Dong, C. Chen, X. Liu, J. Bu, and Y. Gao, "A light-weight and density-aware reprogramming protocol for wireless sensor networks," *IEEE Transactions on Mobile Computing*, no. 99, pp. 1–1, 2011.
- [14] M. Doddavenkatappa, M. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *Proceedings of ACM/USENIX NSDI*, 2013.
- [15] L. Huang and S. Setia, "Cord: energy-efficient reliable bulk data dissemination in sensor networks," in *Proceedings of IEEE INFOCOM*, 2008.
- [16] V. Naik, A. Arora, P. Sinha, and H. Zhang, "Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices," in *Proceedings of IEEE RTSS*, 2005.
- [17] Y. Tseng, S. Ni, and E. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545–557, 2003.
- [18] P. Kyasanur, R. Choudhury, and I. Gupta, "Smart gossip: An adaptive gossip-based broadcasting service for sensor networks," in *Proceedings of IEEE MASS*, 2006.
- [19] W. Lou and J. Wu, "Toward broadcast reliability in mobile ad hoc networks with double coverage," *IEEE Transactions on Mobile Computing*, vol. 6, no. 2, pp. 148–163, 2007.
- [20] Z. Jiang, D. Wu, M. Guo, J. Wu, R. Kline, and X. Wang, "Minimum latency broadcasting with conflict awareness in wireless sensor networks," in *Proceedings of IEEE ICPP*, 2012.
- [21] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, "Synapse++: code dissemination in wireless sensor networks using fountain codes," *IEEE Transactions on Mobile Computing*, vol. 9, no. 12, pp. 1749–1765, 2010.
- [22] D. Starobinski and W. Xiao, "Asymptotically optimal data dissemination in multichannel wireless sensor networks: Single radios suffice," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 695–707, 2010.
- [23] Z. Lu, Y. Shi, W. Wu, and B. Fu, "Data retrieval scheduling for multi-item requests in multi-channel wireless broadcast environments," *IEEE Transactions on Mobile Computing*, vol. 99, no. PrePrints, p. 1, 2013.