Reliable Wireless Broadcasting with Near-Zero Feedback

Weiyao Xiao* Sachin Agarwal† David Starobinski* Ari Trachtenberg*

* Department of Electrical and Computer Engineering
Boston University, Boston, MA 02215, USA
Email:{weiyao,staro,trachten}@bu.edu

† Deutsche Telekom AG, Laboratories
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
Email:sachin.agarwal@telekom.de

***Abstract*—We examine the problem of minimizing feedbacks in reliable wireless broadcasting, by pairing rateless coding with extreme value theory. Our key observation is that, in a broadcast environment, this problem resolves into estimating the *maximum* number of packets dropped among *many* receivers rather than for each individual receiver. With rateless codes, this estimation relates to the number of redundant transmissions needed at the source in order for all receivers to correctly decode a message with high probability. We develop and analyze two new data dissemination protocols, called Random Sampling (*RS*) and Full Sampling with Limited Feedback (*FSLF*), based on the *moment* and *maximum likelihood* estimators in extreme value theory. Both protocols rely on a single-round learning phase, requiring the transmission of a few feedback packets from a small subset of receivers. With fixed overhead, we show that *FSLF* has the desirable property of becoming more accurate as the receivers's population gets larger. Our protocols are channel agnostic, in that they do not require a-priori knowledge of (i.i.d.) packet loss probabilities, which may vary among receivers. We provide simulations and an improved full-scale implementation of the Rateless Deluge over-the-air programming protocol on sensor motes as a demonstration of the practical benefits of our protocols, which translate into about 30% latency and energy consumption savings.**

I. Introduction

Reliable data broadcasting serves as the basis for numerous wireless applications, including over-the-air programming and real-time updating of stock quotes and scoreboards [1–4].

Automatic Repeat reQuest (ARQ) protocols are commonly employed to guarantee the reliability of data dissemination over lossy wireless channels [5]. ARQ requires receivers to notify a source about missing packets via acknowledgements (ACKs) or negative acknowledgements (NACKs). When the number of receivers gets large, however, these messages become excessive and result in the well-known *broadcast storm* problem [2, 4, 6].

Packet-level forward error correction (FEC) provides a promising approach to effectively reduce feedbacks [7]. FEC requires the source to anticipate packet losses and make redundant transmissions proactively, instead of waiting for feedbacks from receivers and then making additional transmissions. Rateless codes such as random linear codes, LT codes [8] and Shifted LT codes [9] allow FEC to be implemented in a practical and efficient way. The source encodes M original packets of a file and then transmit the encoded packets. A receiver is able to recover the file successfully when receiving M (or slightly more) distinct encoded packets.

One of the challenges of implementing FEC is for the source to determine an appropriate amount of redundancy when transmitting proactively. While too many redundant FEC packets slows down the data dissemination process unnecessarily, insufficient redundancy leaves many receivers unable to decode packets. Furthermore, the inherent heterogeneity of channel characteristics across receivers (due to, e.g., link quality, distance to the source, antenna sensitivity) significantly complicates the task of redundancy estimation. While estimating each receiver's packet loss probability may be possible [10], such an approach does not scale given that per-receiver packet loss probability needs to be ascertained.

This paper is based on the following key observation. When using rateless codes in a broadcasting environment, such as wireless, the number of redundant packet transmissions corresponds to the *maximum* number of redundant packet transmissions needed among all receivers. This allows us to exploit advances in extreme value theory [11], a powerful mathematical tool for studying the distribution of extreme order-statistics, such as maxima of random variables, to effectively quantify transmission redundancy with minimum overhead.

We consider the problem of disseminating a file composed of multiple segments, or *pages*, from one source to N receivers over a lossy wireless channel. Each page consists of a fixed number of packets. Our first contribution is formalizing this problem using extreme value theory, in order to perform accurate online estimation of the amount of transmissions (formally defined as δ*-reliable volume* in Section III) a source needs to make in order to achieve a probability δ of successfully delivering each page of the file to all the receivers. Thanks to extreme value theory, we are able to perform accurate estimation of the δ*-reliable volume* without requiring specific knowledge of channel characteristics. This accurate estimation can be accomplished with extrapolation based on limited information obtained from the dissemination of a *single* page.

Second, we develop two new data dissemination protocols,

The work of the first author was performed in part at Deutsche Telekom Laboratories. This work was supported in part by NSF grants CCF-0729158, CCF-0916892 and a grant from Deutsche Telekom Laboratories.

called *Random Sampling* (*RS*) and *Full Sampling with Limited Feedback* (*FSLF*), based on extreme value estimators known to be asymptotically exact as $N \to \infty$. Both protocols estimate the δ-*reliable volume* during a *learning phase*, and then reliably disseminate the rest of the file during a *transmission phase*. While *RS* restricts the overhead of the estimation during the *learning phase*, by randomly sampling feedbacks from a small subset of receivers, *FSLF* judiciously exploits the fact that the extreme value estimators require only samples of the $k+1$ largest order statistics, for some $k << N$, to collect all the feedbacks needed. We further show that *FSLF* has the appealing property of providing more accurate estimation of the δ-*reliable volume* when the receivers' population gets larger, without incurring higher overhead.

Third, we show through extensive simulations that both *RS* and *FSLF* almost completely eliminate receivers' feedbacks during the *transmission phase*. Thanks to the high accuracy of the estimators, the amount of packet transmission by the source is only about 5% higher than with ARQ.

Fourth, we compare the performance of different extreme value estimators, namely, the *moment* and *maximum likelihood* estimators, in conjunction with the *RS* and *FSLF* protocols. Amongst all the possible combinations, we observe that *FSLF* based on the moment estimator achieves the best performance, in terms of minimizing overhead and maximizing accuracy.

Finally, to demonstrate practical benefits of our protocols, we conduct *real mote* experiments on a testbed of 14 Tmote Sky sensors [12] and perform larger-scale simulation using the TOSSIM simulator [13]. Specifically, we design a new over-the-air programming protocol based on Rateless Deluge [4], called Extreme Value Quantile Estimation (EV-QE) Deluge, which integrates the *RS* protocol. The experiments and simulations show that EV-QE Deluge lead to a 75% reduction in control-plane traffic together with 30% savings on latency and energy consumption, at the expense of an about 5% increase in data plane traffic with respect to Rateless Deluge.

This paper is organized as follows. In Section II we survey related work. We formulate our problem in Section III. We point out the limitations of classical estimation techniques in Section IV-A, give a primer on extreme value theory in Section IV-B, and introduce the moment and maximum likelihood estimators in Section IV-C. The design of the *RS* and *FSLF* protocols is presented in Section V. Simulation results and sensor mote experiments are provided in Section VI and VII, respectively. We provide concluding remarks in Section VIII.

II. Related Work

The concept of exploiting FEC for reliable data dissemination has been the subject of prior research, both in wireline and wireless settings, and we next survey those work most related to our paper. The works in [14, 15] numerically evaluate the performance improvements achieved with different levels of FEC redundancy. In order to allow the sender to decide when to stop transmitting FEC-coded packets without explicit feedback, the authors in [16–19] study the properties of file dissemination completion times. While in practice the packet loss probability differs from node to node due to many factors (i.e., link quality, distance to the source, antenna sensitivity), the works in [14, 16, 20, 21] as well as some hybrid FEC/ARQ protocols such as [22], assume homogeneous packet loss probabilities in their analysis. The work in [15, 17–19] do study the more realistic scenario of heterogeneous packet loss, but they assume that receivers' packet loss probabilities are *known* to the source, a relatively strong assumption for practical multiple receiver environments. In our work, we allow the packet loss probabilities to be *unknown* and *heterogeneous* across receivers.

While it is possible to perform online estimation of network parameters such as packet loss probabilities [10], such techniques are not generally scalable with the number of receivers in the network, given that all per-receiver packet loss probabilities must be determined. The authors in [23] try to estimate FEC redundancy without obtaining the individual receivers' packet loss probabilities, but they do not establish relationship between the redundancy and the probability of success. In this present work, we propose to estimate the amount of transmissions needed to fully disseminate data to all receivers, with probability δ. Our estimate is computed online without the knowledge of channel characteristics, and we establish an analytical relationship between the amount of transmissions and the probability of success.

Finally, estimation can also be performed using classical approaches [24]. However, they have significant overhead (cf. Section IV for details), which our approach avoids by utilizing the theory of extreme values.

III. Problem Formulation

We consider the problem of broadcasting a file from a source (e.g., base station) to N receivers within its transmission range. The file is divided into R pages, each consisting of M packets. Encoding is done at the packet level using rateless codes (e.g. computing random sums of input packets). Each receiver needs to receive M distinct packets[1] in order to recover a page.

The time axis is slotted, and each packet transmission is assumed to take one time slot. The packet loss probability for receiver n $(n = 1, .., N)$ is p_n, where p_n's are *heterogenous* and *unknown*, but assumed to be independent, identically distributed (i.i.d.) random variables. The source encodes and broadcasts the pages in an increasing order. Sending one page is denoted as one *realization* in the data dissemination process.

In a given realization r of the data dissemination process, denote by T_n^r the number of time slots required for receiver n to recover a page. Since p_n's are i.i.d. random variables, the time required for decoding the page is i.i.d. across receivers; in other words, T_n^rs are also i.i.d. random variables. Denote by T^r the random variable representing the completion time for this realization, *i.e.*,the number of time slots needed to disseminate M packets to a cluster of N receivers: $T^r = \max_{n=1...N} T_n^r$.

The *success probability* of a page dissemination is the probability that the page is decoded by all N receivers. The

[1] Some *near optimal* rateless codes require slightly more than M packets to decode a page.

δ-reliable volume, denoted as t_δ, is the amount of packets the source needs to broadcast to guarantee a success probability δ.

Our goal is to sample and analyze a fixed number of feedback packets in a single realization, corresponding to the broadcast of the first page of a file, in order to estimate the *δ-reliable volume* t_δ. Our estimation aims to accurately quantify the value t_δ of a realization r, where, by definition, $\Pr\{T^r \leq t_\delta\} = \delta$. Note that t_δ is also referred to as *δ-quantile* of the distribution function $\Pr\{T^r \leq t\}$ [25, p.404].

IV. Extreme Value Quantile Estimation

We begin this section with a review of traditional approaches in quantile estimation, pointing out their limitations, and a short primer on extreme value theory, a powerful statistic tool for studying the distribution of the maxima of random variables. We then introduce extreme value theory-based estimators, which form the basis for our near-zero feedback data dissemination protocols described in Section V.

As discussed in Section III, for a given realization r of the data dissemination process, the completion times of the receivers are i.i.d. random variables, $T^r_1, T^r_2, .., T^r_N$, following an unknown distribution function $F(t)$. Let their order statistics be $T^r_{1,N}, T^r_{2,N}, .., T^r_{N,N}$, meaning that $T^r_{1,N} \leq T^r_{2,N} \leq .. \leq T^r_{N,N}$. Clearly, $T^r_{N,N}$, which corresponds to the maximum of completion times among all receivers, is identical to T^r, the number of packet transmissions by the source during realization r. Similarly, for R realizations, $r = 1, .., R$, the set of T^r's are also i.i.d. random variables because they are the maxima of i.i.d. random variables. Let their order statistics be denoted by $T^{1,R}, T^{2,R}, .., T^{R,R}$.

Recall that our goal is to quantify the *δ-reliable volume*, t_δ, needed in order to achieve a success probability δ of delivering a page to all receivers, corresponding to the δ-quantile of the distribution function $\Pr\{T^r \leq t\} = \Pr\{\max_{n=1,..N} T^r_n \leq t\} = F^N(t)$ [25, p.404]. Equivalently, this problem can be considered as estimating τ-quantile, $\tau = \delta^{\frac{1}{N}}$, of the distribution function $F(t)$; clearly this τ-quantile is precisely t_δ.

A. Classical Estimators and their Limitations

Classical quantile estimators compute the δ-quantile, $\widehat{t}_\delta$, by interpolating linearly between the order statistics [25, p.404]. For example, consider the averaging quantile estimator [24]. After the completion times of R realizations are collected and ordered as $T^{1,R} \leq T^{2,R} \leq \ldots \leq T^{R,R}$, the δ-quantile for $\Pr\{T^r \leq t\} = F^N(t)$ is estimated as

$$\widehat{t}_\delta = \begin{cases} \frac{1}{2}(T^{j,R} + T^{j+1,R}) & \text{if } j = N\delta, \\ T^{j+1,R} & \text{otherwise.} \end{cases} \tag{1}$$

A major limitation of all interpolation-based quantile estimators is that they need many realizations (*i.e.*,large R) to estimate high quantiles. The fundamental reason is that all these estimators implicitly assume that the estimation will not exceed the largest order statistic, namely, $T^{R,R}$. For instance, using Eq. (1) to estimate the high quantile (when $\delta > 1 - \frac{1}{R}$) always yields $\widehat{t}_\delta = T^{R,R}$. Therefore, this estimator becomes ineffective when $\delta > 1 - \frac{1}{R}$. In other words, it is not possible to estimate any quantile higher than $(1 - \frac{1}{R})$ based on the data collected from R realizations using classical quantile estimators. Equivalently, to determine t_δ where $\Pr\{T^r \leq t_\delta\} = \delta$, one needs to collect the completion times of T^rs from at least $R = \frac{1}{1-\delta}$ realizations.

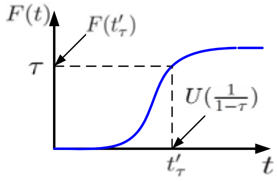


Fig. 1. The relationship between function F and U.

Note that one could equivalently estimate the τ-quantile, $\tau = \delta^{\frac{1}{N}}$, of the distribution function $F(t)$ by collecting the individual completion times, T^r_n, from all receivers. However, it can be shown that at least $R = \frac{1}{1-\delta}$ realizations are still required for classical quantile estimators.

B. Extreme Value Theory

The completion time for successfully disseminating a page corresponds to the maximum of the individual completion times of each receiver. In order to estimate the *δ-reliable volume* by extrapolating beyond the limited amount of feedbacks (based on only a single realization), one needs to explore the properties of the distribution of the maximum of i.i.d. random variables.

Extreme Value Theory (EVT) provides a sound theoretical framework for such an extrapolation. It restricts the behavior of the distribution of the maximum of i.i.d. random variables, namely T^r, where $T^r = \max_{n=1,..N} T^r_n$, to an EVT distribution. The EVT distribution can be specified by just two parameters, the *extreme value index* and the *scale factor* [11]. Consequently, we can quantify the *δ-reliable volume* without requiring knowledge of channel statistics of each individual receiver.

Formally, suppose there exists a sequence of constants $a(N)$ and $b(N)$, such that $(\max_{n=1,..N} T^r_n - b(N))/a(N)$ has a non-degenerate limit distribution as $N \to \infty$. Then, according to extreme value theory,

$$\lim_{N\to\infty} F^N(a(N)t + b(N)) = G_\gamma(t), \tag{2}$$

where

$$G_\gamma(t) = \exp\left(-(1+\gamma t)^{-\frac{1}{\gamma}}\right), \text{ for } 1+\gamma t > 0, \gamma \in \mathbb{R}, \tag{3}$$

and the right-hand side interpreted as $\exp(-e^{-t})$ for $\gamma = 0$.

Define U to be the inverse function of $\frac{1}{1-F}$. As depicted in Fig. 1, $U(\frac{1}{1-\tau})$ corresponds to τ-quantile, t'_τ. Ref. [11] shows that the following statement is equivalent to Eq. (2):

$$\lim_{\frac{N}{k}\to\infty} \frac{U(t) - U(\frac{N}{k})}{a(\frac{N}{k})} = \frac{(t\frac{N}{k})^\gamma - 1}{\gamma}, \tag{4}$$

where $U(N) = b(N)$ in Eq. (2), and the right-hand side is interpreted as $\log t$ for $\gamma = 0$.

Eq. (4) is used as a basis for extreme quantile estimation. To estimate a τ-quantile, one can use following estimator,

$$\widehat{t}_\tau = \widehat{U}(\frac{1}{1-\tau}) = \widehat{U}(\frac{N}{k}) + \widehat{a}(\frac{N}{k})\frac{(\frac{1}{1-\tau}\cdot\frac{k}{N})^{\widehat{\gamma}}-1}{\widehat{\gamma}}, \quad (5)$$

where N is the *sample size* and k is the *intermediate number*. As $k \to \infty$, $N \to \infty$ and $\frac{k}{N} \to 0$, Eq. (5) asymptotically converges to the actual quantile [11].

By the definition of U, it can be shown that $\widehat{U}(\frac{N}{k}) = T^r_{N-k,N}$. The right hand side, $T^r_{N-k,N}$, which is the $(N-k)$ largest completion time in the r-th realization, can be obtained from the order statistics of the empirically collected completion times reported by the receivers. Therefore, when using Eq. (5) to estimate the τ-quantile, one only needs to estimate the *extreme value index* γ and the *scale factor* $a(\frac{N}{k})$. This is the reason the δ*-reliable volume* at the source can be estimated without knowledge of channel characteristics. Next, we describe statistical approaches for estimating the two parameters.

C. Estimation of the Extreme Value Index and Scale Factor

We will now introduce two important EVT estimators used to estimate the extreme value index γ and the scale factor $a(\frac{N}{k})$ of Eq. (5). Note that these estimators are derived from Eq. (2) or its equivalent forms.

1) The Moment Estimator [26]: The *moment estimator* is an extension of the simple and widely used Hill estimator [27], which is a special case $j = 1$ of the following equation:

$$M_N^{(j)} = \frac{1}{k}\sum_{i=0}^{k-1}(\log T^r_{N-i,N} - \log T^r_{N-k,N})^j, j = 1, 2. \quad (6)$$

The Hill's estimator provides estimates for $\gamma_+ \triangleq \max(0,\gamma)$, and converges to 0 when $\gamma < 0$ (*i.e.,*non-informative).

Let $\gamma_- \triangleq \min(0,\gamma)$. The work [26] estimates γ_- as follows:

$$\widehat{\gamma}_- = 1 - \frac{1}{2}\left(1 - \frac{(M_N^{(1)})^2}{M_N^{(2)}}\right)^{-1}. \quad (7)$$

Complementarily to the Hill's estimator, $\widehat{\gamma}_-$ can only estimate the case where $\gamma < 0$ and converges to 0 for the case $\gamma \geq 0$.

The moment estimator for $\gamma \in \mathbb{R}$, proposed in [26], is essentially a combination of the estimator for γ_+ and γ_-,

$$\widehat{\gamma}_M = M_N^{(1)} + 1 - \frac{1}{2}\left(1 - \frac{(M_N^{(1)})^2}{M_N^{(2)}}\right)^{-1}. \quad (8)$$

The corresponding moment estimator of the scale factor is

$$\widehat{a}_M(\frac{N}{k}) = X_{N-k,N}M_N^{(1)}(1 - \widehat{\gamma}_-). \quad (9)$$

The authors in [26] show that provided $k = k(N) \to \infty$ and $\frac{k}{N} \to 0$, as $N \to \infty$, both $\widehat{\gamma}_M$ and $\widehat{a}_M$ are consistent estimators and converge to γ and $a(\frac{N}{k})$.

2) The Maximum Likelihood Estimator [28]: Given a set of observations $T^r_1, .., T^r_N$, the *maximum likelihood* (ML) estimator aims to determine which parameters of the extreme distribution make the observed data most likely to occur. We next summarize the work in [28], which provides an equivalent method of approximating Eq. (2).

Denote the upper endpoint of F by $t^* = \sup\{t : F(t) < 1\} \leq \infty$ and, for $s < t^*$, let $F_s(t)$ be the conditional distribution function of $T^r_n - s$ given $T^r_n > s$. More precisely,

$$F_s(t) = P(T^r_n \leq t + s | T^r_n > s) = \frac{F(s+t) - F(s)}{1 - F(s)}, \quad (10)$$

for $s < t^*$, $t > 0$ and $1 - F(s) > 0$.

Then, (based on [28] and the citations therein) there exists a normalizing function $a(s) > 0$ such that

$$\lim_{s \to t^*} \sup_{0<t<t^*-s} \left| F_s(t) - H_\gamma(\frac{t}{a(s)}) \right| = 0, \quad (11)$$

if and only if F is in the maximum domain of attraction of $G_\gamma(t)$, and $H_\gamma(t)$ is the generalized Pareto distribution function

$$H_\gamma(t) = 1 - (1 + \gamma t)^{-\frac{1}{\gamma}}. \quad (12)$$

Eq. (11) shows that the distribution of an applicable random variable $T - s$ given $T > s$ converges to a generalized Pareto distribution $H_\gamma(t)$, as $s \to t^*$. This suggests that, given a set of order statistics of random variables $T^r_{1,N}, .., T^r_{N,N}$, the distribution of the set of random variables $\left\{(T^r_{N-k+i,N} - T^r_{N-k,N}) | i = 1 \ldots k\right\}$ given $T^r_{N-k,N}$, can be approximated by the distribution of an ordered sample of k i.i.d. random variables with CDF $H_\gamma(\frac{t}{a(s)})$, where $s = T^r_{N-k,N}$.

Given k observations $(T^r_{N-k+1,N} - T^r_{N-k,N}), .., (T^r_{N,N} - T^r_{N-k,N})$, we may thus obtain an approximative likelihood function, $\sum_{i=1}^k \log h_{\gamma,a(s)}(T^r_{N-i+1,N} - T^r_{N-k,N})$, where $h_{\gamma,a(s)}(t) = \frac{\partial H_\gamma(t/a(s))}{\partial t}$ is the PDF (Probability Density Function) of $H_\gamma(\frac{t}{a(s)})$.

In order to obtain the estimation that maximizes the likelihood, one can set the partial derivatives of the likelihood function with respect to the extreme value index γ and the scale function $a(s)$ to zero. Therefore, the resulting likelihood system of equations in terms of the excesses $T^r_{N-i+1,N} - T^r_{N-k,N}$, $i = 1, .., k$ are as follows:

$$\left\{\begin{array}{l} \displaystyle\sum_{i=1}^{k}\frac{1}{\gamma^2}\log\left(1 + \frac{\gamma}{a(\frac{N}{k})}(T^r_{N-i+1,N} - T^r_{N-k,N})\right) \\ \displaystyle -\sum_{i=1}^{k}\left(\frac{1}{\gamma}+1\right)\frac{\frac{1}{a(\frac{N}{k})}(T^r_{N-i+1,N} - T^r_{N-k,N})}{1 + \frac{\gamma}{a(\frac{N}{k})}(T^r_{N-i+1,N} - T^r_{N-k,N})} = 0, \\ \displaystyle\sum_{i=1}^{k}\left(\frac{1}{\gamma}+1\right)\frac{\frac{\gamma}{a(\frac{N}{k})}(T^r_{N-i+1,N} - T^r_{N-k,N})}{1 + \frac{\gamma}{a(\frac{N}{k})}(T^r_{N-i+1,N} - T^r_{N-k,N})} = k. \end{array}\right. \quad (13)$$

There are only two unknown variables in Eq. (13), which are γ and $a(\frac{N}{k})$, and their solutions are the ML estimators for extreme value index and scale factor, denoted $\widehat{\gamma}_{MLE}$ and $\widehat{a}_{MLE}(\frac{N}{k})$ respectively. Discussions on obtaining the solutions of Eq. (13) numerically can be found in [29].

Under certain technical conditions [28], for $k = o(\log^2 N) \to \infty$ and $N \to \infty$, $\widehat{\gamma}_{MLE}$ and $\widehat{a}_{MLE}$ are shown to asymptotically converge to the actual values, γ and $a(\frac{N}{k})$.

Remarks:

1) Both the moment and ML estimators need *only* the largest $k+1$ order statistics (of N samples) to estimate γ and $a(\frac{N}{k})$, a significant source of savings for our protocols.

2) According to [11], different choices of the intermediate number k result in different estimation variances. Discussion on tuning the parameters of the estimators is beyond the scope of this paper due to space constraints, and we refer the interested reader to our technical report in [30].

3) It is possible to use the EVT estimators to collect data from multiple realizations (pages) to improve the estimation quality. However, we show that the estimation has small variance (see Section VI) even if the collection is restricted to one realization.

V. Broadcasting Protocols with Limited Feedbacks

In practical applications, one of the crucial steps to start the estimation is to collect sample data, which is referred to the *learning phase* in this paper. The estimation will then be used to determine the required redundancy for the remaining pages, which are distributed in the *transmission phase* of our protocols.

In the learning phase, the source disseminates the first page to the network and then collects individual completion times (T_n^rs) from the receivers. Upon collecting enough responses, the source estimates δ*-reliable volume* using either the moment estimator based on Eq. (8) and Eq. (9), or the ML estimator based on Eq. (13). The estimate of t_δ is used to determine how many packets to transmit in the transmission phase.

It is important to minimize the communication overhead of our protocol (in terms of the duration and the number of feedbacks) in order to maintain scalability, especially when there are a large number of receivers. We propose two methods for managing this overhead - *Random Sampling* (*RS*) and *Full Sampling with Limited Feedback* (*FSLF*).

A. Random Sampling

1) Learning Phase: In our first approach, *Random Sampling*, the source restricts the amount of feedbacks by only collecting completion times from a small set (N') of receivers chosen uniformly at random among N receivers, where $N' \leq N$.

In order to collect feedbacks from N' random receivers, the source attaches a random seed to the first page, and then encodes and transmits packets as usual. The common seed is used by all receivers to generate the same set of N' pseudo-random integers uniformly distributed on $[1 \ldots N]$. Only receivers with IDs within this common set send their completion time back to the source. The source continues to send packets until it receives N' feedbacks. It then uses the feedbacks to estimate t_δ. In practice, the feedback channel itself may be faulty, in which case one may designate some smaller threshold of feedback packets that must be received before attempting estimation.

As discussed in Section IV, to quantify t_δ, one can either estimate the δ-quantile of $\Pr\{T_r \leq t\} = F^N(t)$, or estimate the τ-quantile, $\tau = \delta^{\frac{1}{N}}$ of $\Pr\{T_r^n \leq t\} = F(t)$. In our case, we will estimate the τ-quantile, since the source collects completion times T_n^r during the dissemination process.

According to Eq. (5), with N' data samples, $T_{1,N'}^r, .., T_{N',N'}^r$, the source first sorts the data and then obtains the number of transmissions required using following estimator:

$$\widehat{T}_{RS}(\delta) = \widehat{U}\left(\frac{N'}{k}\right) + \widehat{a}\left(\frac{N'}{k}\right) \frac{\left(\frac{1}{1-\delta^{\frac{1}{N}}} \cdot \frac{k}{N'}\right)^{\widehat{\gamma}} - 1}{\widehat{\gamma}}. \quad (14)$$

The result will be one of two different δ*-reliable volume* estimators, $\widehat{T}_{RS}^{M}(\delta)$ corresponding the moment estimator or $\widehat{T}_{RS}^{MLE}(\delta)$, corresponding to the ML estimator.

We will show through simulation in Section VI-A that randomly choosing $N' = 50$ out of $N = 10^4$ receivers is sufficient to achieve good estimation.

2) Transmission Phase: The source first broadcasts data packets as estimated in the learning phase. Receivers that cannot recover the page, sense the channel and, if no other request is overheard first, reply to the source with a request for additional data packets. The source transmits $2^{i-1}\eta$ additional packets in the i-th round of its transmissions, where η is an integer. This multiplicative factor is important in reducing the number of requests, and our simulations show that it does not typically result in transmission of unnecessary data packets. If there is no request for a predefined length of time T_{report}, the source moves on to the next page. In our sensor-mote implementation (Section VII), we used $\eta = 1$ and $T_{report} = 500$ (ms).

B. Full Sampling with Limited Feedbacks

A simple way to improve the quality of estimations using the *RS* is to collect more feedbacks, e.g., increasing N' towards N. However, this approach does not scale as the number of receivers N becomes large and may cause feedback implosion, precisely the problem we try to avoid in the first place.

By exploiting the inherent properties of the EVT estimators, we devise here a sampling approach called *Full Sampling with Limited Feedbacks* which is able to *essentially* collect completion times from all receivers, with almost fixed amount of feedbacks. Consequently, an appealing property of *FSLF* is that, given a fixed amount of feedbacks, the estimators become more accurate when the network has more receivers, since the source collects more useful data samples.

FSLF exploits the fact that the EVT estimators only need the $k+1$ (recalled k is the intermediate number) largest order statistics $T_{N-k,N}^r, T_{N-k+1,N}^r, .., T_{N,N}^r$, as inputs for the estimation. Therefore, if the sorting process can be performed before the collecting process, and the source only collects the $k+1$ largest individual completion times from the network, then this is equivalent to the case where the source collects all the data, sorts them, and then use the $k+1$ largest order statistics as inputs for the extreme estimators to quantify δ*-reliable volume*.

In order for the source to collect the $k+1$ largest completion times, we assign higher priority to the receivers with larger completion times in sending feedbacks. This is achieved as follow. The source transmits the first page using the *RS* method. Ideally, after the page is successfully disseminated, all receivers know the network completion time $T^r = \max_{n=1,..N} T_n^r$. Each receiver then sets a random timer with length inversely proportional to the difference between its own completion time T_n^r and the network completion time T^r. Before the timer expires, each receiver records the number of overheard feedbacks with completion time larger than or equal to its own. When the timer expires, a receiver reports its own completion time T_n^r if less than $k+1$ feedbacks are recorded, or suppress it otherwise.

The timer of each receiver is set as follows. Recall T_{report} is the time allowed for receivers to report the feedbacks. We set the length of the timer to be a random variable uniformly distributed in the interval between $\frac{T_{report}}{T^r-M+1}(T^r - T_n^r)$ and $\frac{T_{report}}{T^r-M+1}(T^r - T_n^r + 1)$. Therefore, a receiver with larger individual completion time T_n^r will report its time sooner. After waiting for T_{report} of time, the source then estimates *δ-reliable volume* with the $k+1$ largest order statistics.

In practice, each receiver may not necessarily know the network completion time and the source may not be able to collect all $k+1$ largest completion times, due to lossy channels. We let each receiver consider the time when overhearing the last data packets as the network completion time, and use it in lieu of T^r. In the case where the source collects k' ($k' < k+1$) feedbacks, the source will underestimate t_δ since it considers the k' feedbacks as the largest k' completion times. However, recall the source already has an estimation from *RS* when sending the first page. Therefore, it can compare both estimates and keep the larger one.

The estimators for *FSLF* are slightly different from *RS*. With *FSLF*, although the source collects only $k+1$ completion times, it is equivalent to the case where it collects the completion times from all the receivers, sorts them and then use the $k+1$ largest order statistics as inputs for the extreme estimators. Thus, the sample size for *FSLF* is N. One can first create N data by setting the $k+1$ largest ones to the samples collected, and the rest to zero, then obtain the estimation of t_δ as following,

$$\widehat{T}_{FSLF}(\delta) = \widehat{U}(\frac{N}{k}) + \widehat{a}(\frac{N}{k})\frac{(\frac{1}{1-\delta^{\frac{1}{N}}}\cdot\frac{k}{N})^{\widehat{\gamma}}-1}{\widehat{\gamma}}. \qquad (15)$$

We again have two different *δ-reliable volume* estimators, $\widehat{T}_{FSLF}^{M}(\delta)$ and $\widehat{T}_{FSLF}^{MLE}(\delta)$, depending on the estimators used (moment or ML). After the estimation, the system enters the transmission phase and behaves the same way as described in the *RS* method.

In summary, an importantly property of *FSLF* is that given the same amount of feedbacks, it achieves higher accuracy as the number of the receivers becomes larger. *FSLF* helps to mitigate the problem of feedback implosion in the learning phase, as it restricts the number of feedbacks to be close to $k+1$. Therefore, thanks to its scalability and increasing accuracy, this approach is ideal for broadcasting in dense networks.

C. Overhead Analysis of Extreme Value Estimators

We next summarize the overhead of the EVT estimators and compare it with that of classical approaches. We look at the number of feedbacks needed for the estimation as well as the number of pages, which corresponds to the time needed.

According to the discussion in Section IV-A, to estimate t_δ, a classical estimator needs to know the completion time of $\frac{1}{1-\delta}$ pages to get a valid estimation. Therefore, the learning phase of classical estimators requires the transmission of at least $\frac{1}{1-\delta}$ pages and the collection of the completion time for each page.

For EVT estimators, the learning phase for both *RS* and *FSLF* requires the transmission of only one page to estimate t_δ. During the learning phase of *RS*, only N' receivers report their completion times. Therefore the number of feedbacks needed for *RS* estimation is N'. In the learning phase of *FSLF*, the source transmits the first page using *RS*, and then collects the $k+1$ largest completion times. Therefore, the number of feedbacks needed for *FSLF* is $N'+k+1$.

Note that these comparisons are for the best case of all estimators. In practice, the difference between them can be even larger as shown by our simulations and experiments.

VI. Numerical Results

A. Performance of Extreme Value Estimators

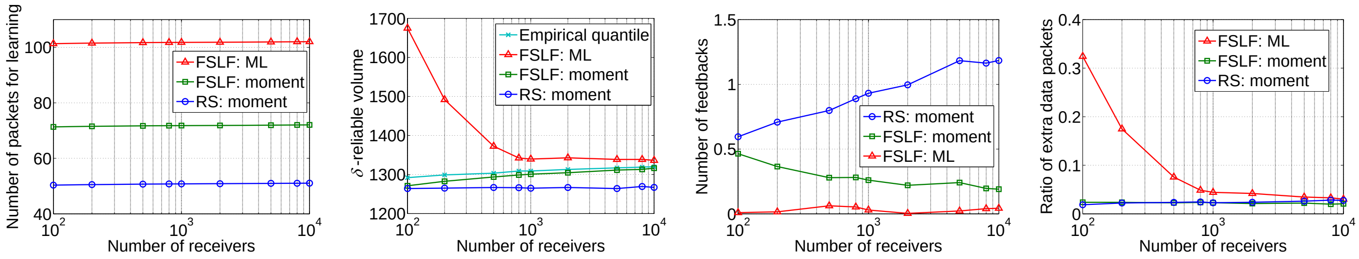
We first investigate the overhead and accuracy of the EVT estimators proposed in Section V in the *learning phase*, as well as the benefit of applying the estimation to the *transmission phase*, in terms of reducing feedback requests and maintaining the minimum required *δ-reliable volume*, t_δ, where $\delta = 99\%$.

In this simulation, a two-page file is disseminated to N (ranging from 10^2 to 10^4) receivers. Each page consists of $M = 1000$ packets. For receiver n, the corresponding packet loss rate p_n, *unknown* to the source, is a uniformly distributed random variable in the range $[0, 0.2]$. The *δ-reliable volume* estimations are obtained from Eq. (14) and Eq. (15) for *RS* and *FSLF*, respectively. The extreme value index and scale factor are estimated by the moment estimator (Eq. (8) and Eq. (9)), and the ML estimator (by solving Eq. (13) using Matlab).

For *RS*, the source collects feedbacks from $N' = 50$ random receivers. The intermediate number is $k = 20$. Since the solution of the system of equations for the ML estimator yields complex solutions when N' is small (similar issue is reported in [11]), we omit the ML estimation here. For *FSLF*, the value of k for the moment estimator and the ML estimator is set to 20 and 50, respectively. The results shown in the following figures represent an average over 1000 iterations.

Fig. 2(a) shows that the overhead of the estimators (i.e., the number of packets collected for the estimation) marginally increases as the number of receivers N grows. As discussed in Section V-C, the smallest possible overhead for *RS* and *FSLF* to perform estimation is N' and $N'+k+1$ feedbacks, respectively. The result shows that the overhead for both estimators is close to minimum and remains almost a constant as the number of receivers increases. Note that the overhead of *FSLF* is slightly higher than *RS*, since *FSLF* needs to collect the $k+1$ largest completion times at the end of the learning phase. Next we will show that this extra communication cost trades off with higher accuracy in estimating the *δ-reliable volume*.

Fig. 2(b) shows the accuracy of the estimators by comparing the estimations with the empirical quantile. The empirical quantile is obtained from the classical quantile estimator in Eq. (1)



(a) (b) (c) (d)

Fig. 2. Performance of EVT estimators. (a) Learning phase: the overhead of quantile estimation, (b) Learning phase: the accuracy of the estimators, (c) Transmission phase: the amount of feedback, (d) Transmission phase: the ratio of extra packets sent to the minimum packets needed.

by running 10^5 identical iterations, bringing this estimate close to the actual value. The accuracy of the *RS* approach decreases as the number of receivers increases. Recall that estimating δ-quantile, t_δ, of $F^N(t)$ is equivalent to estimating the τ-quantile, t'_τ, $\tau = \delta^{\frac{1}{N}}$ of $F(t)$ (see Section IV). Therefore, increasing the number of receivers requires greater extrapolation to estimate higher quantiles. Since *RS* fixes the number of feedbacks to $N' = 50$, its accuracy thus decreases.

On the other hand, the accuracy of *FSLF*-based estimators improves as the number of receivers grows, because the source collects more useful data with increasing N. Correspondingly, the EVT estimates converge to the actual value as $N \to \infty$. Further, the overhead of *FSLF* barely increases as N grows, as shown in Fig. 2(a), confirming the scalability of *FSLF*.

Next, we study the benefit of applying the estimation in reducing control traffic during the transmission phase. We record (1), the number of feedback requests and (2), the extra number of data packets transmitted compared to a pure ARQ scheme, where the source only transmits when requested and so the number of data packets transmitted is minimum.

Fig. 2(c) shows the average number of feedbacks for encoded packets during the transmission phase. While transmitting a page with $M = 1000$ packets to $N = 10^4$ receivers using *RS* with moment estimator, the average number of requests is only 1.2 per page. It even decreases when using *FSLF*, 0.19 for the moment estimator and 0.04 for the ML estimator. Therefore, in almost all cases, receivers recover the page using the initially estimated *δ-reliable volume* without using feedbacks.

Fig. 2(d) shows that the moment estimator using both *RS* and *FSLF* overestimates the required redundancy by only 4% (of the total number of packets transmitted). In the ML estimator, the overestimate is relatively large when the number of receivers is small, but decreases to a reasonable level for larger N.

Based on the above simulations, we believe that *FLSF* using the moment estimator provides the best trade-off along all dimensions of interest (high estimation accuracy and low number of feedbacks). *RS* using the moment estimator is accurate when the number of receivers is small. However, the amount of feedbacks required needs to grow with increasing N to maintain accurate estimation. Both protocols drastically reduce receiver(s)-to-sender traffic and incur only marginal extra communication due to overestimating t_δ.

B. *Small Network and Non-i.i.d. Scenarios*

Here we evaluate the estimators under a non-typical EVT scenario, i.e., a network with small number of receivers that have non-i.i.d. packet loss rates. We consider a two state burst channel model in Fig. 3(a). When the receivers are in a good channel state their packet loss rates are heterogeneous random variables uniformly distributed in the range $[0, 0.2]$. When there is error-burst (with probability $P_{G,B} = 0.2$), the channel will switch to the bad state, wherein the packet loss of each receiver is uniformly distributed in the range $[0.6, 0.8]$. The channel may switch back to the good state with probability $P_{B,G} = 0.5$ during a subsequent time slot.

Similar to previous simulations, the source first transmits one page, collects feedbacks, and estimates t_δ. It then transmits the next pages based on the estimate. The parameters are as follows, $M = 50$, N ranging from 20 to 60, $N' = 20$, $k = 10$, and $\delta = 95\%$. The results in Fig. 3(b) and Fig. 3(c) demonstrate that for these scenarios, the estimators still significantly reduce feedbacks from the receivers. The extra communication due to overestimating the *δ reliable volume* is slightly larger than in i.i.d. packet loss scenarios, but it is reasonable.

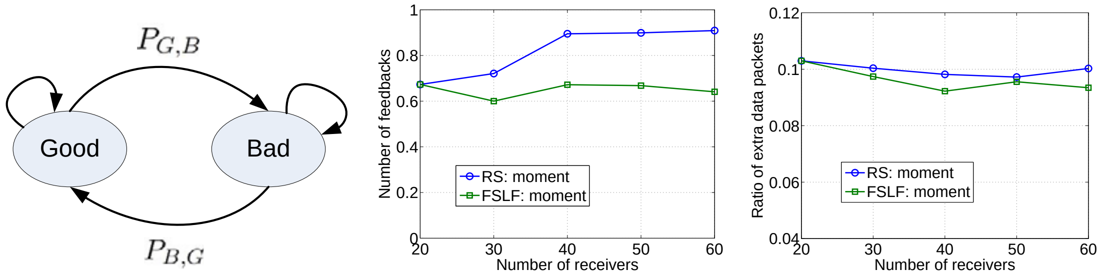
VII. Prototype Implementation

In this section, we enhance an over-the-air programming protocol for wireless sensor networks using the proposed extreme value techniques. Our modifications are based on the Rateless Deluge over-the-air programming protocol [4], which uses random linear codes for efficient file distribution to wireless sensors. The performance of both protocols is compared using our Tmote sky [12] testbed as well as through the TOSSIM bit-level network simulator [13].

A. *Setup*

In our setup a file is divided into pages consisting of twenty 23-byte packets each. The packet loss rate of each receiver is a uniform random variable in the range [0.1, 0.2]. All sensors are within communication range and transmit at their highest power setting to ensure a good link, and packet loss at the receiver is forced by dropping packets uniformly at random according to its own packet loss rate. All results in this section represent an average of 10 independent trials.

A sensor requests encoded packets from the sender if it discovers that its neighbors have new data. The request message



(a) (b) (c)

Fig. 3. Extreme value estimators for network with small number of receivers with non-i.i.d. packet loss rates. (a) Two state burst channel model, (b) The amount of feedback, (c) The amount of extra data communication.

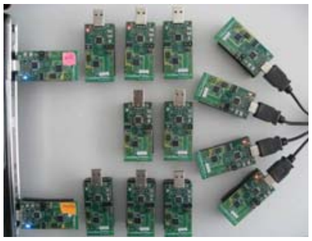
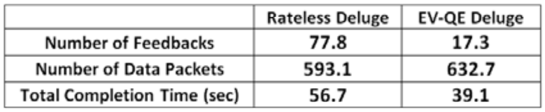


Fig. 4. Tmote Sky sensors testbed with 14 motes.



	Rateless Deluge	EV-QE Deluge
Number of Feedbacks	77.8	17.3
Number of Data Packets	593.1	632.7
Total Completion Time (sec)	56.7	39.1

Fig. 5. Rateless Deluge vs. EV-QE Deluge: 20 pages, $N = 12$, $M = 20$, heterogeneous packet loss.

specifies the page number and the number of packets needed. When a sensor receives enough packets, it can decode the page successfully. A sensor suppresses its request if it has overheard similar requests by other sensors recently.

Here, we augment the original Rateless Deluge with the extreme value quantile estimation technique, and refer to the new protocol as EV-QE Deluge. To ensure a fair comparison, minimal modifications are made to Rateless Deluge. EV-QE Deluge operates in the same manner as Rateless Deluge when disseminating the first page, referred to the *learning phase* in Section V. The source then uses the *RS* approach to collect N' random feedbacks from the receivers and estimate the δ-*reliable volume* corresponding to success probability $\delta = 0.95$.

In the *transmission phase*, the source initially disseminates a page based on the estimated δ-*reliable volume*. After that, it waits for a certain amount of time ($T_{report} = 512$ (ms)). In case a receiver requests more encoded packets in this time, the requested encoded packets are transmitted. Otherwise the source continues to send the next page.

B. Tmote Sky Sensor Testbed

The performance of EV-QE Deluge and the original Rateless Deluge is first evaluated on a testbed consisting of 14 Tmote Sky sensors. One sensor serves as the file-sending base station and 12 other sensors are receivers. The last sensor is used to record network traffic.

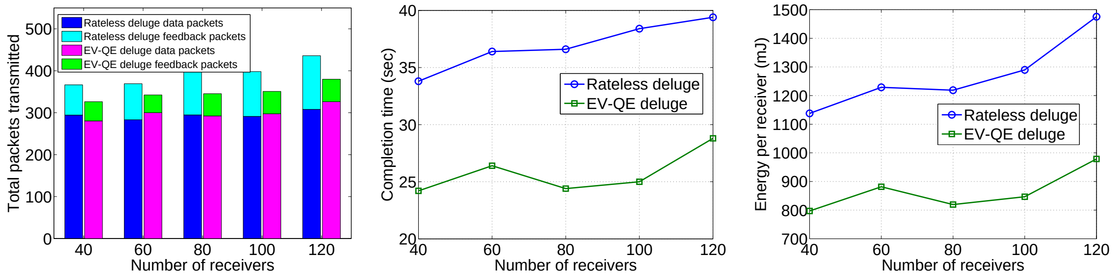
The size of the file is 8,518 bytes, which corresponds to 20 pages using Rateless Deluge and EV-QE Deluge. We monitor the network traffic due to the encoded packets transmitted and due to the encoded packet requests. We also record the overall completion time of disseminating the file. Since the number of receivers here is small (12 sensor motes), we have the source collect the feedbacks from every receiver after disseminating the first page. Namely, $N' = N = 12$ in the first experiment. The intermediate number k is set to 5.

The results in Fig. 5 show that EV-QE Deluge sends out slightly more encoded packets (about 6%). However, it drastically reduces the amount of feedbacks, which is only 17.3 on average. Note that this number includes the overhead messages in the learning phase for the estimation of t_δ, which is 12, as well as the request messages when the source transmits the first page using the original Rateless Deluge, which is 3.9 on average, as shown in Fig. 5. Therefore, with EV-QE Deluge, in the transmission phase, the average number of feedbacks is about 1.4 for 19 pages in total, indicating that most of the time the entire network finishes receiving enough packets after the source's first set of transmissions for each page. Being able to accurately estimate t_δ, EV-QE Deluge effectively reduces the overall data dissemination time by about 30%.

C. Large Scale Network Simulation with TOSSIM

We next compare the performance of both protocols in TOSSIM for a larger scale experiment. The energy consumption (due to CPU and Radio) for both protocols are also monitored through PowerTOSSIM [31]. The parameters for the *RS* method are set to $N' = 30$, $k = 20$.

The simulation results for varying number of receivers N, are shown in Fig. 6(a), 6(b) and 6(c). As expected, the number of data packets sent out by EV-QE Deluge is slightly higher than Rateless Deluge. However, as N increases, the number of feedbacks of EV-QE Deluge remains almost constant at about 50, including the $N' = 30$ initial feedbacks for the source to estimate δ-*reliable volume*. On the other hand, the number of feedbacks of Rateless Deluge increases with N. By reducing the control overhead, EV-QE Deluge is able to effectively reduce the overall completion time and energy consumption per



(a) (b) (c)

Fig. 6. Rateless Deluge vs. EV-QE Deluge, TOSSIM simulation: 9 pages, $M = 20$, heterogeneous packet loss, varying the number of receivers N. (a) Total packets transmitted: forward and feedback channels, (b) Completion time, (c) Average energy consumption per receiver.

receiver by about 30%, as shown in Fig. 6(b), and Fig. 6(c).

VIII. Concluding Remarks

In this paper, we propose novel, on-line prediction mechanisms for FEC-coded data dissemination in wireless networks with heterogenous packet loss probabilities. Our mechanisms, based on the (asymptotically exact) moment and ML estimators in extreme value theory, offer major scalability benefits because (1) estimation of per-receiver packet loss probabilities is not required; (2) the number of feedbacks used to estimate redundancy is nearly constant; (3) accuracy improves with growth in the number of receivers.

We introduce two new protocols, *RS* and *FSLF*, for wireless data broadcasting. Our simulation results show that the *FSLF* protocol, in conjunction with the moment estimator, provides a good trade-off between the number of feedbacks used to estimate redundancy, the redundancy transmitted, and the number of extra encoded packet transmission requests. Further, we verify that our approach provides reasonable performance even when the assumptions of i.i.d. packet loss probability and large number of receivers are relaxed.

We demonstrate practical feasibility of our proposed approach by integrating *RS* into the Rateless Deluge OAP protocol on a testbed of T-sky sensor motes. Our experimental and simulation results indicate a 30% reduction in latency and energy consumption, an improvement of particular significance for battery-limited wireless devices.

References

[1] J. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale." in *SenSys'04*, Nov. 2004.

[2] W. Xiao and D. Starobinski, "Poster abstract: Exploiting multi-channel diversity to speed up over-the-air programming of wireless sensor networks," in *SenSys'05*, San Diego, California, USA, Nov. 2005.

[3] D. Starobinski, W. Xiao, X. Qin, and A. Trachtenberg, "Near-optimal data dissemination policies for multi-channel, single radio wireless sensor networks," in *IEEE INFOCOM 2007*, Anchorage, May 2007.

[4] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes," in *IPSN 2008*, Saint Louis, MO, USA, Apr. 2008.

[5] P. Levis, N. Patel, S. Shenker, and D. Culler, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," University of California at Berkeley, Tech. Rep., 2004.

[6] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 375–386, 1999.

[7] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," *INFOCOM '90, IEEE*, Jun 1990.

[8] M. Luby, "Lt codes," in *FOCS '02*, Washington, DC, 2002, pp. 271–280.

[9] A. Hagedorn, S. Agarwal, D. Starobinski, and A. Trachtenberg, "Rateless coding with feedback," in *In proc. IEEE INFOCOM*, April 2009.

[10] D. He, N. Akhtar, R. Tafazolli, A.-L. Burness, and P. Eardley, "On-line packet loss rate estimation," *PIMRC'05*, vol. 3, pp. 1570–1574, 2005.

[11] L. de Haan and A. Ferreira, *Extreme Value Theory, An Introduction*. Springer, 2006.

[12] http://www.moteiv.com/.

[13] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *SenSys '03*, 2003.

[14] C. Huitema, "The case for packet level FEC," in *Protocols for High-Speed Networks'96*. Chapman & Hall, Ltd., 1996, pp. 109–120.

[15] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," in *SIGCOMM '97*, 1997.

[16] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," *INFOCOM 2008.*, April 2008.

[17] M. Mosko and J. J. Garcia-Luna-Aceves, "An analysis of packet loss correlation in FEC-enhanced multicast trees," in *ICNP '00*, 2000.

[18] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On delay performance gains from network coding," *CISS*, 2006.

[19] W. Xiao and D. Starobinski, "Extreme value FEC for wireless data broadcasting," in *IEEE INFOCOM '09*, Rio de Janeiro, Brazil, April 2009.

[20] M. Ghaderi, D. Towsley, and J. Kurose, "Network coding performance for reliable multicast," *MILCOM 2007. IEEE*, pp. 1–7, Oct. 2007.

[21] Y. Bartal, J. Byers, M. Luby, and D. Raz, "Feedback-free multicast prefix protocols," *ISCC '98.*, pp. 135–141, 30 Jun.-2 Jul. 1998.

[22] L. Rizzo and L. Vicisano, "RMDP: an FEC-based reliable multicast protocol for wireless environments," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 2, no. 2, pp. 23–31, 1998.

[23] S. Fahmy and R. M. Jegadeesan, "Recovery Performance of PGM with Redundancy Estimation for Proactive FEC," *Technical report CSD-TR-03-019, Purdue University*, 2003.

[24] T. Dielman, C. Lowry, and R. Pfaffenberger, "A comparison of quantile estimators," *Communications in Statistics - Simulation and Computation*, vol. 23, no. 2, pp. 355 – 371, 1994.

[25] M. J. Schervish, *Theory of Statistics*. Springer, 1996.

[26] A. L. M. Dekkers and J. H. J. Einmahl and L. De Haan, "A Moment Estimator for the Index of an Extreme-Value Distribution," *Ann. Statist.*, vol. 17, no. 4, pp. 1833–1855, 1989.

[27] B. M. Hill, "A Simple General Approach to Inference About the Tail of a Distribution," *Ann. Statist.*, vol. 3, no. 5, pp. 1163–1174, 1975.

[28] H. Drees and A. Ferreira and L. de Haan, "On maximum likelihood estimation of the extreme value index," *Ann. Appl. Probab.*, vol. 14, no. 3, pp. 1179–1201, 2004.

[29] S. D. Grimshaw, "Computing Maximum Likelihood Estimates for the Generalized Pareto Distribution," *Technometrics*, vol. 35, no. 2, pp. 185–191, 1993.

[30] W. Xiao, S. Agarwal, D. Starobinski, and A. Trachtenberg, "Reliable Wireless Broadcasting with Near-Zero Feedback," in *CISE Technical Report: 2009-IR-0040*, Boston University, 2009.

[31] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *SenSys '04*, 2004, pp. 188–200.