Reliable Rateless 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 feedback 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' 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. Further, we apply our protocols to real time oblivious (RT) rateless codes in broadcast settings. Through simulations, we demonstrate a 100-fold reduction in the amount of feedback packets while incurring an increase of only 10-20% in the number of encoded packets transmissions.**

***Index Terms*—Extreme Value Theory, Forward Error Correction (FEC), Rateless Coding, Over-the-Air Programming.**

I. Introduction

Reliable data broadcasting is the basis for over-the-air programming (OAP) of sensor networks [1–4]. OAP is used to deliver software updates and data from a broadcaster (source) to large populations of sensors (receivers) within wireless transmission range of the source. Similar protocols have also been developed for applications like real-time updating of stock-quotes and score-boards on cellular and mobile smartphones.

A preliminary version of this paper appeared in the proceedings of the IEEE INFOCOM 2010 conference.

This work was supported in part by NSF grants CCF-0729158, CCF-0916892 and a grant from Deutsche Telekom Laboratories.

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 acknowledgments (ACKs) or negative acknowledgments (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 feedback [7]. FEC requires the source to anticipate packet losses and make redundant transmissions proactively, instead of waiting for feedback 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 transmits the encoded packets. A receiver is able to recover the file successfully after 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 (e.g., due to link quality, distance to the source, and 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.

In this paper, motivated by OAP applications, 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, 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 feedback 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 feedback 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. These results for *FSLF* hold under the assumption that receivers can overhear each others' transmissions.

Third, we show through extensive simulations that both *RS* and *FSLF* almost completely eliminate receivers' feedback 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.

Fifth, 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 as compared to Rateless Deluge.

Finally, we employ extreme value estimators for creating scalable broadcast versions of real time oblivious (RT) rateless codes [14]. RT codes have a simple decoder design, making them especially suitable for wireless sensor receivers. However, each receiver is required to send information about its decoding progress to the source periodically, limiting the scalability of RT codes for large receiver populations. For a marginal increase in the number of encoded packets broadcast from the source, we show how using EVT to predict the decoding progress offers a major decrease (on the order of two orders of magnitude) in the total amount of feedback required from the receivers.

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 illustrate the practical applicability of our approach by designing and then benchmarking a scalable broadcast version of RT codes [14] in Section VIII. We provide concluding remarks in Section IX.

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 works most related to our paper. The works in [15, 16] 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 [17–20] 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 [15, 17, 21, 22] as well as some hybrid FEC/ARQ protocols such as [23], assume homogeneous packet loss probabilities in their analysis. The works in [16, 18–20] 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. The idea of applying EVT to minimize feedback was first proposed in [20]. However, the techniques presented in this present paper are completely different from those in [20], since we resort here to *on-line measurements* to estimate the extreme-value parameters.

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 [24] 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 [25]. 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., a 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). Our analysis and simulations in Sections IV, V, and VI assume idealized rateless codes, where each receiver needs to receive precisely M distinct packets in order to recover a page. Sections VII and VIII show how to apply the results to some practical rateless codes, namely random linear codes and RT codes.

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 *heterogeneous* 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 the packet losses are governed by i.i.d. random variables, the time required for decoding the page is i.i.d. across receivers; in other words, T_n^r's 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 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, $\Pr\{T^r \le t_\delta\} = \delta$. Note that t_δ is also referred to as δ-*quantile* of the distribution function $\Pr\{T^r \le t\}$ [26, 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 statistical tool for studying the distribution of the maxima of random variables. We then introduce extreme value theory-based estimators, which form the basis of 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, $T_1^r, T_2^r, .., T_N^r$, following an unknown distribution function $F(t)$. Let their order statistics be $T_{1,N}^r, T_{2,N}^r, .., T_{N,N}^r$, meaning that $T_{1,N}^r \le T_{2,N}^r \le .. \le T_{N,N}^r$. Clearly, $T_{N,N}^r$, 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. T_n^r. 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 $\Pr\{T^r \le t\} = \Pr\{\max_{n=1,..N} T_n^r \le t\} = F^N(t)$ [26, p.404]. Equivalently, this problem can be considered as estimating the τ-quantile $\tau = \delta^{\frac{1}{N}}$ of the distribution function $F(t)$. The 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 [26, p.404]. For example, consider the averaging quantile estimator [25]. After the completion times of R realizations are collected and ordered as $T^{1,R} \le T^{2,R} \le \ldots \le T^{R,R}$, the δ-quantile for $\Pr\{T^r \le t\} = F^R(t)$ is estimated as

$$\widehat{t_\delta} = \begin{cases} \frac{1}{2}(T^{j,R} + T^{j+1,R}) & \text{if } \delta = \frac{j}{R}, j = 1, ..R-1 \\ T^{j+1,R} & \text{if } \frac{j}{R} < \delta < \frac{j+1}{R}, j = 0, ..R-1. \end{cases} \tag{1}$$

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 $\Pr\{T^r \le t_\delta\} = \delta$, one needs to collect the completion times of T^rs from at least $R = \frac{1}{1-\delta}$ realizations.

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_n^r, from all receivers. However, it can be shown that at least $R = \frac{1}{1-\delta}$ realizations are still required.

B. Extreme Value Theory

The completion time for successfully disseminating a page corresponds to the maximum of the individual completion times

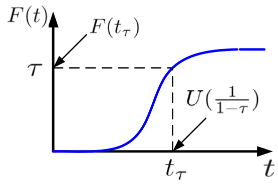


Fig. 1. The relationship between function F and U.

of each receiver. In order to estimate the δ-*reliable volume* by extrapolating beyond the limited amount of feedback (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 $T^r = \max\limits_{n=1,..N} T^r_n$, to an EVT distribution. The EVT distribution can be specified by just two parameters, the *extreme value index* and the *scale factor* [11], defined below. Consequently, we can quantify the δ-*reliable volume* without requiring knowledge of channel statistics of each individual receiver. Formally:

Theorem 1 ([11], Theorem 1.1.3): Suppose there exists a sequence of constants $a(N)$ and $b(N)$, such that $\frac{\max_{n=1,..N} T^r_n - b(N)}{a(N)}$ has a non-degenerate limit distribution as $N \to \infty$. Then

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

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 is interpreted to be $\exp(-e^{-t})$ when $\gamma = 0$.

Define U to be the inverse function of $\frac{1}{1-F}$. As depicted in Fig. 1, $U(\frac{1}{1-\tau})$ corresponds to the τ-quantile t_τ of $F(t)$. The function U is more convenient to work with when performing quantile estimation. The next theorem relates the asymptotic behavior of U to that of F.

Theorem 2 ([11], Theorem 1.1.6): The following statement is equivalent to Eq. (2). There exists a positive function a, such that for $x > 0$

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

and the right-hand side is interpreted to be $\log x$ when $\gamma = 0$. Moreover, Eq. (2) holds with $b(N) = U(N)$ and the same function a.

Eq. (4) can be used as the basis for extreme quantile estimation. Let N be the *sample size* and k be the *intermediate number*, where, as $N \to \infty$, $k \to \infty$, and $\frac{k}{N} \to 0$. Then, one can use the following estimator for the τ-quantile (see [11, page 67]):

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

From the definition of U, it can be shown that $U(x) = F^{-1}(1-\frac{1}{x})$. Since $F^{-1}(T^r_{N-k,N})$ corresponds to the $\frac{N-k}{N}$-quantile for $k < N$, we have

$$\widehat{U}(\frac{N}{k}) = T^r_{N-k,N}, \tag{6}$$

which is the $(N-k)$ largest completion time in the r-th realization. This quantity is readily obtained by computing the order statistics of the empirical 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 why the δ-*reliable volume* at the source can be estimated without knowledge of channel characteristics. Next, we describe statistical approaches for estimating these 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 [11, 27]: The *moment estimator* is an extension of the simple and widely used Hill estimator [28], which is a special case $j = 1$ of the following equation:

$$M_N^{(j)} = \frac{1}{k}\sum_{i=0}^{k-1}(\log T^r_{N-i,N} - \log T^r_{N-k,N})^j, j = 1, 2. \tag{7}$$

The Hill's estimator provides an estimate for $\gamma_+ \triangleq \max(0, \gamma)$ (*i.e.*, $\widehat{\gamma}_+ = M_N^{(1)}$). Thus, $\widehat{\gamma}_+ \to 0$ when $\gamma < 0$ (*i.e.*, it is non-informative). Let $\gamma_- \triangleq \min(0, \gamma)$. This quantity can be estimated as follows:

$$\widehat{\gamma}_- = 1 - \frac{1}{2}\left(1 - \frac{(M_N^{(1)})^2}{M_N^{(2)}}\right)^{-1}. \tag{8}$$

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 $\gamma \in \mathbb{R}$ is a combination of the estimator for γ_+ and γ_-

$$\widehat{\gamma}_M = M_N^{(1)} + 1 - \frac{1}{2}\left(1 - \frac{(M_N^{(1)})^2}{M_N^{(2)}}\right)^{-1}. \tag{9}$$

The corresponding moment estimator of the scale factor is

$$\widehat{a}_M(\frac{N}{k}) = T_{N-k,N} M_N^{(1)}(1 - \widehat{\gamma}_-). \tag{10}$$

The following theorem states that $\widehat{\gamma}_M$ and $\widehat{a}_M(\frac{N}{k})$ are *consistent* estimators (*i.e.*, they converge in probability to the actual values of γ and $a(\frac{N}{k})$). In the following, we denote the upper endpoint of $F(t)$ by $t^* = \sup\{t : F(t) < 1\} \leq \infty$ and use the notation $\to_p$ to denote convergence in probability.

Theorem 3 ([11], Theorems 3.5.2 and 4.2.1): Suppose Eq. (2) holds and $t^* > 0$. Let $\widehat{\gamma}_M$ and $\widehat{a}_M(\frac{N}{k})$ be defined as in Eq. (9) and Eq. (10). Then

$$\widehat{\gamma}_M \to_p \gamma \qquad \text{and} \qquad \frac{\widehat{a}_M(\frac{N}{k})}{a(\frac{N}{k})} \to_p 1, \tag{11}$$

provided $k = k(N) \to \infty$ and $\frac{k}{N} \to 0$, as $N \to \infty$.

From the definitions of $\widehat{U}(\frac{N}{k})$, $\widehat{\gamma}_M$, $\widehat{a}_M(\frac{N}{k})$ in Eq. (6), Eq. (9), and Eq. (10) respectively, we deduce the following important result, namely that only the $k+1$ largest order statistics (of the N samples) are needed to compute the estimator of the τ-quantile given by Eq. (4).

Corollary 1: Collecting the $k+1$ largest order statistics of T_n^r (*i.e.*, $T_{N-K,N}^r, T_{N-K+1,N}^r, \ldots, T_{N,N}^r$) is sufficient for the computation of $\widehat{t}_\tau$.

2) The Maximum Likelihood Estimator [29]: Given a set of observations $T_{1,N}^r, \ldots, T_{N,N}^r$, 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 [29], which provides an equivalent method of approximating Eq. (5).

As before, the upper endpoint of F is denoted $t^* = \sup\{t : F(t) < 1\} \leq \infty$. For $s < t^*$, let $F_s(t)$ be the conditional distribution function of $T_n^r - s$ given $T_n^r > s$. More precisely,

$$F_s(t) = P(T_n^r \leq t + s | T_n^r > s) = \frac{F(s+t) - F(s)}{1 - F(s)}, \tag{12}$$

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

Let $H_\gamma(t)$ be the generalized Pareto distribution function

$$H_\gamma(t) = 1 - (1 + \gamma t)^{-\frac{1}{\gamma}}. \tag{13}$$

Then, (based on [11, 29] and the citations therein) there exists a normalizing function $a(s) > 0$, which is the same as that defined in Eq. (4), such that

$$\lim_{s \to t^*} \sup_{0<t<t^*-s} \left| F_s(t) - H_\gamma(\frac{t}{a(s)}) \right| = 0, \tag{14}$$

if and only if F_s is in the maximum domain of attraction of $G_\gamma(t)$.

Eq. (14) shows that the distribution of an applicable random variable $T - s$ given $T > s$ converges to a generalized Pareto distribution $H_\gamma(t)$, as $s \to t^*$. Therefore, $H_\gamma(\frac{t}{a(s)})$, which is determined by the parameters γ and $a(s)$, can be used to approximate $F_s(t)$.

The ML estimator aims to determine the parameters which make the observed data most likely to occur [30]. Specifically, given a set of $\mathcal{L}$ independent observations $t_1, t_2, .., t_{\mathcal{L}}$ (drawn from $H_\gamma(\frac{t}{a(s)})$), the ML estimator determines values of γ and $a(s)$ that maximize the joint probability that these observations will occur. Formally, let $h_{\gamma,a(s)}(t) = \frac{\partial H_\gamma(t/a(s))}{\partial t}$ be the PDF (Probability Density Function) of $H_\gamma(\frac{t}{a(s)})$. Thus we have

$$h_{\gamma,a(s)}(t) = \frac{1}{a(s)} \left(1 + \gamma \frac{t}{a(s)}\right)^{-\frac{1}{\gamma}-1}. \tag{15}$$

Therefore, the joint density function for all $\mathcal{L}$ independent observations is as follows,

$$h_{\gamma,a(s)}(t_1, t_2, .., t_{\mathcal{L}}) = \Pi_{i=1}^{\mathcal{L}} h_{\gamma,a(s)}(t_i). \tag{16}$$

Eq. (16) is also called *likelihood* function. The goal of the ML estimator is to find the values of γ and $a(s)$ that maximize the likelihood when given the observations $t_1, t_2, .., t_{\mathcal{L}}$. Namely,

$$\{\widehat{\gamma}_{MLE}, \widehat{a}(s)_{MLE}\} = \operatorname*{argmax}_{\gamma, a(s)} \Pi_{i=1}^{\mathcal{L}} h_{\gamma,a(s)}(t_i). \tag{17}$$

Equivalently, one can maximize the logarithm of the likelihood function, called *log-likelihood*, as following,

$$\{\widehat{\gamma}_{MLE}, \widehat{a}(s)_{MLE}\} = \operatorname*{argmax}_{\gamma, a(s)} \sum_{i=1}^{\mathcal{L}} \log h_{\gamma,a(s)}(t_i). \tag{18}$$

In order to obtain the estimation that maximizes the likelihood, one can set the partial derivatives of the log-likelihood function (Eq. (18)) with respect to γ and $a(s)$ to zero. Therefore, using Eq. (15), one can obtain the ML estimation for γ and $a(s)$ by solving the following system of equations,

$$\begin{cases} \dfrac{\partial \log h_{\gamma,a(s)}(t)}{\partial \gamma} = 0, \\ \dfrac{\partial \log h_{\gamma,a(s)}(t)}{\partial a(s)} = 0. \end{cases} \tag{19}$$

Namely,

$$\begin{cases} \displaystyle\sum_{i=1}^{\mathcal{L}} \frac{1}{\gamma^2} \log\left(1 + \frac{\gamma}{a(s)}(t_i)\right) \\ \displaystyle -\sum_{i=1}^{\mathcal{L}} \left(\frac{1}{\gamma} + 1\right) \frac{\frac{t_i}{a(s)}}{1 + \frac{\gamma}{a(s)} t_i} = 0, \\ \displaystyle -\sum_{i=1}^{\mathcal{L}} \frac{1}{a(s)} + \sum_{i=1}^{\mathcal{L}} \left(\frac{1}{\gamma} + 1\right) \frac{\frac{\gamma}{a^2(s)} t_i}{1 + \frac{\gamma}{a(s)} t_i} = 0. \end{cases} \tag{20}$$

Now back to the problem of estimating the extreme value index and the scale factor using ML estimator. When given a set of order statistics of random variables $T_{1,N}^r, .., T_{N,N}^r$, the distribution of the set of random variables $\left\{(T_{N-k+i,N}^r - T_{N-k,N}^r) | i = 1 \ldots k\right\}$ given $T_{N-k,N}^r$, $F_s(t)$, 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 $s = T_{N-k,N}^r$. According to Eq. (20), one thus can obtain the ML estimators for the extreme value index γ and the scale factor $a(\frac{N}{k})$ by solving the following system of equations,

$$\begin{cases} \displaystyle\sum_{i=1}^{k} \frac{1}{\gamma^2} \log\left(1 + \frac{\gamma}{a(\frac{N}{k})}(T_{N-i+1,N}^r - T_{N-k,N}^r)\right) \\ \displaystyle -\sum_{i=1}^{k} \left(\frac{1}{\gamma} + 1\right) \frac{\frac{1}{a(\frac{N}{k})}(T_{N-i+1,N}^r - T_{N-k,N}^r)}{1 + \frac{\gamma}{a(\frac{N}{k})}(T_{N-i+1,N}^r - T_{N-k,N}^r)} = 0, \\ \displaystyle\sum_{i=1}^{k} \left(\frac{1}{\gamma} + 1\right) \frac{\frac{\gamma}{a(\frac{N}{k})}(T_{N-i+1,N}^r - T_{N-k,N}^r)}{1 + \frac{\gamma}{a(\frac{N}{k})}(T_{N-i+1,N}^r - T_{N-k,N}^r)} = k. \end{cases} \tag{21}$$

Algorithm 1 Learning Phase: Random Sampling (RS) at the source

```
Attach a common random seed to each packet in the first page
repeat
    Broadcast a new encoded packet
    if any completion times received from receivers then
        store received completion times
    end if
until N' completion times are collected
Perform estimation
```

There are only two unknown variables in Eq. (21), which are γ and $a(\frac{N}{k})$, and their solutions are the ML estimators for the 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. (21) numerically can be found in [31].

Under appropriate technical conditions [29], it can be shown that $\widehat{\gamma}_{MLE} - \gamma$ and $\widehat{a}_{MLE}(\frac{N}{k}) - a(\frac{N}{k})$ converge to asymptotic Normal distributions, under suitable normalization. Furthermore, similar to the moment estimators, the ML estimators only require knowledge of the $k+1$ largest order statistics.

V. Broadcasting Protocols with Limited Feedback

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 a subset of the receivers. Upon collecting enough responses, the source estimates the *δ-reliable volume* using either the moment estimator based on Eq. (9) and Eq. (10), or the ML estimator based on Eq. (21). 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 amount of feedback) in order to maintain scalability, especially when there is 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 feedback by only collecting completion times from a small subset (N') of receivers chosen uniformly at random among N receivers, where $N' \le N$. 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.

In order to collect feedback from N' random receivers, the source keeps encoding and transmitting packets till it receives feedback from N' receivers with their completion time. It also attaches a common random seed to each packet. The source's algorithm in the learning phase is shown in Algorithm 1.

Algorithm 2 Learning Phase: RS at the receivers

```
Keep receiving encoded packets until the first page is successfully decoded
Use the common random seed to generate a set of N' different pseudo-random integers uniformly distributed on [1...N]
if node ID belongs to the set of N' integers then
    send completion time to the source
end if
```

After decoding the first page, 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' feedback packets. It then uses the feedback to estimate t_δ. The receivers' algorithm in the learning phase is shown in Algorithm 2. In practice the feedback channel itself may be faulty, or a receiver with ID among the N' chosen receivers may fail to decode the first page. To deal with both of these issues, one may elect to use a smaller threshold on the number of feedback packets that must be received before attempting estimation at the source.

As discussed in Section IV, to quantify t_δ, one can either estimate the δ-quantile of $\Pr\{T_r \le t\} = F^N(t)$, or estimate the τ-quantile, $\tau = \delta^{\frac{1}{N}}$ of $\Pr\{T_r^n \le t\} = F(t)$. 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 the following estimator:

$$\widehat{T}_{RS}(\delta) = \widehat{U}\left(\frac{N'}{k}\right) + \widehat{a}\left(\frac{N'}{k}\right) \frac{\left(\frac{1}{1-\delta^{\frac{1}{N}}} \cdot \frac{k}{N'}\right)^{\widehat{\gamma}} - 1}{\widehat{\gamma}}. \quad (22)$$

The result will be one of two different *δ-reliable volume* estimators, $\widehat{T}_{RS}^{M}(\delta)$ corresponding to the moment estimator or $\widehat{T}_{RS}^{MLE}(\delta)$, corresponding to the ML estimator. Theorem 3 implies that the moment estimators of γ and $a(\frac{N}{k})$ are consistent as $N' \to N$ and $N \to \infty$.

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 rounds (i.e., making it proportional to the logarithm of the number of additional packet transmissions needed). Our simulations indicate that this approach does not typically result in many unnecessary transmissions. If there is no request for a

Algorithm 3 Transmission Phase: RS at the source

repeat
  Encode a new page
  Broadcast the required number of encoded packets (determined by estimation)
  $i \leftarrow 1$
  repeat
    Broadcast $2^{i-1}\eta$ packets
    $i = i + 1$;
  until no request for additional encoded packets is received within T_{report} amount of time
until no new page to broadcast

Algorithm 4 Transmission Phase: RS at the receivers

repeat
  Decode a new page
  repeat
    Receive new encoded packets
    if encoded broadcast packets no longer received **then**
      Request encoded packets from source
    end if
    Try to decode the page
  until successfully decode the current page
until all pages are decoded

predefined length of time T_{report}, the source moves on to the next page. In our sensor mote implementation (Section VII), we use $\eta = 1$ and $T_{report} = 500$ (ms). Algorithm 3 and 4 show the algorithms respectively run by the source and the receivers during the transmission phase.

B. Full Sampling with Limited Feedback

A simple way to improve the quality of estimations using the *RS* is to collect more feedback, e.g., by increasing N' toward 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 a sampling approach called *Full Sampling with Limited Feedback* which is able to collect all the completion times needed from the receivers, with an almost fixed amount of feedback. Consequently, an appealing property of *FSLF* is that, given a fixed amount of feedback, the estimators become more accurate when the network has more receivers, since the source collects more useful data samples. *FSLF* is designed for the case where receivers can hear each other.

Although *FSLF* broadly mirrors the Algorithms 1 and 2 during the learning phase, the approach for collecting samples is different. It exploits the fact that the EVT estimators only need the $k + 1$ (recall k is the intermediate number) largest order statistics $T^r_{N-k,N}, T^r_{N-k+1,N}, .., T^r_{N,N}$ as inputs for the estimation. Therefore, if the sorting process can be performed before the collection process, and the source only collects the

Algorithm 5 Learning Phase: FSLF, at receiver n

Keep receiving encoded packets until the first page is successfully decoded and record time T^r_n
Listen to the channel until last packet is sent by the source and record time T^r
Wait for a random time period proportional to $T^r - T^r_n$, and record the number of feedback packets containing receiver completion times sent to the source during that period
if number of feedback packets transmitted by other receivers to source $\leq k$ **then**
  Send T^r_n to source
end if

$k+1$ largest individual completion times from all the receivers in the network, then this is equivalent to the case where the source collects all the data, sorts them, and then uses the $k+1$ largest order statistics as inputs for the extreme estimators to quantify the δ*-reliable volume*.

In order for the source to collect the $k+1$ largest completion times, receivers with larger completion times are granted higher priority in sending feedback. This is achieved as illustrated in Algorithm 5. The source transmits the first page as in the *RS* protocol. Ideally, after the page is successfully disseminated, all receivers record the network completion time $T^r = \max_{n=1,..N} T^r_n$. Each receiver then sets a random timer with length inversely proportional to the difference between its own completion time T^r_n and the network completion time T^r (note that only the difference between T^r and T^r_n is needed, not the absolute value of these variables). Before the timer expires, each receiver records the number of overheard feedback packets with completion time larger than or equal to its own. When the timer expires, a receiver reports its own completion time T^r_n only if less than $k + 1$ feedback packets have been recorded.

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

In practice, each receiver may not precisely 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 packet sent by the source as the network completion time, and use it in lieu of T^r. In the case where the source collects $k' < k + 1$ feedback packets, it may consider them as the k' largest completion times. In such a case, the source may underestimate t_δ. However, the source also 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, one obtains the estimation of t_δ as following,

$$\widehat{T}_{FSLF}(\delta) = \widehat{U}(\frac{N}{k}) + \widehat{a}(\frac{N}{k})\frac{(\frac{1}{1-\delta^{\frac{1}{N}}} \cdot \frac{k}{N})^{\widehat{\gamma}} - 1}{\widehat{\gamma}}. \quad (23)$$

We again have two different δ-*reliable volume* estimators, $\widehat{T}^{M}_{FSLF}(\delta)$ and $\widehat{T}^{MLE}_{FSLF}(\delta)$, depending on the estimators used (moment or ML). Further, assuming that the source received the $k+1$ largest completion times, Corollary 1 implies that the moment estimators of γ and $a(\frac{N}{k})$ are consistent as $N \to \infty$.

In summary, an important property of *FSLF* is that given the same amount of feedback, 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 feedback packets. 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 feedback packets 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 feedback packets 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 feedback packets 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. (22) and Eq. (23) for *RS* and *FSLF*, respectively. The extreme value index and scale factor are estimated by the moment estimator (Eq. (9) and Eq. (10)), and the ML estimator (by solving Eq. (21) using Matlab).

For *RS*, the source collects feedback 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 values of k for the moment estimator and the ML estimator are 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 $N'+k+1$ feedback packets, 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. Since a small intermediate number k for the ML estimator yields to complex solutions, it is set to 50 for the ML estimator, larger than the one for the moment estimator, which is 20. Recall for the *FSLF* sampling technique, the intermediate number corresponds to the number of samples the source needs to collect from the receivers. Therefore, *FSLF* with the ML estimator has higher overhead than *FSLF* with the moment estimator in the learning phase, as shown in Fig. 2(a). 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) 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 feedback 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.

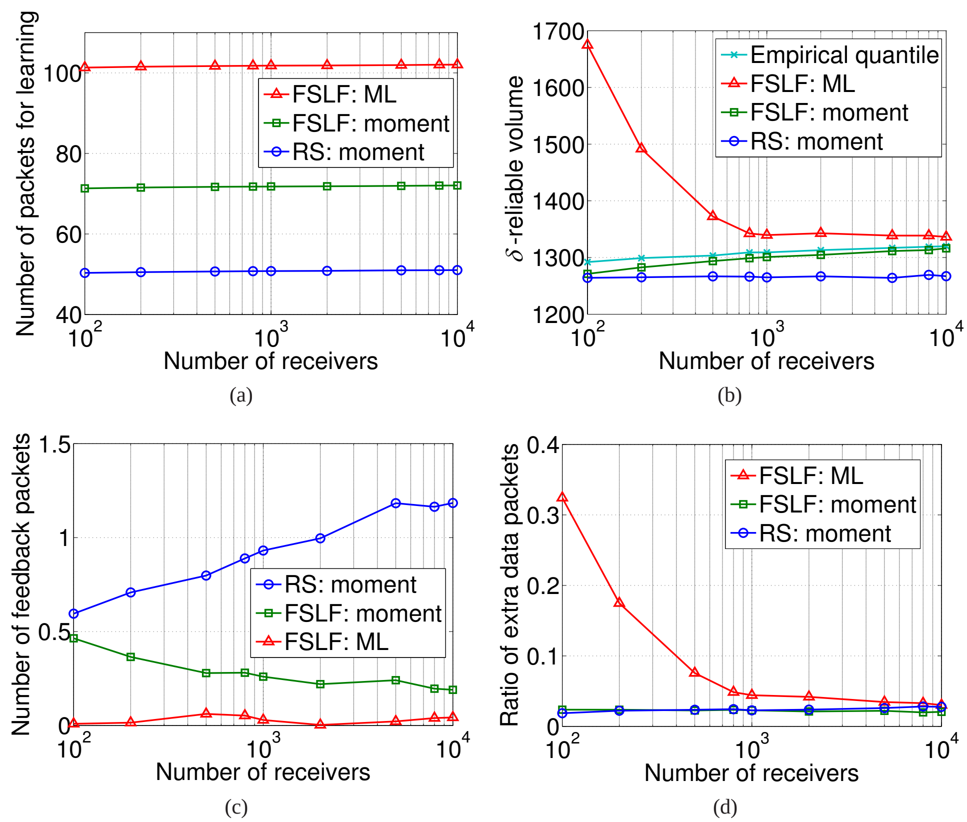


Fig. 2. Performance of EVT estimators. (a) Learning phase: the overhead of quantile estimation, (b) Learning phase: the accuracy of the estimators, (c) Transmission phase: the amount of feedback, (d) Transmission phase: the ratio of extra packets sent to the minimum packets needed.

Fig. 2(c) shows the average number of feedback packets during the transmission phase. When transmitting a page of $M = 1000$ packets to $N = 10^4$ receivers using the *RS* and moment estimator methods, the average number of feedback packets is only 1.2 per page. When using *FSLF*, it decreases to 0.19 for the moment estimator and 0.04 for the ML estimator. Therefore, in all cases, receivers recover the page using the initially estimated δ*-reliable volume* with little or no feedback.

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 conclude that *FLSF* using the moment estimator provides the best trade-off along all dimensions of interest (high estimation accuracy and low amount of feedback). *RS* using the moment estimator is accurate when the number of receivers is small. However, the amount of feedback 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.

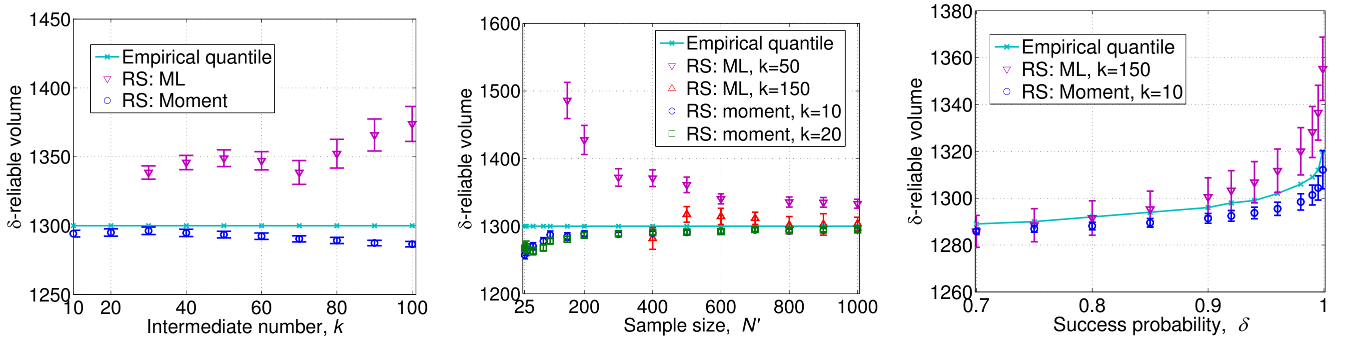
B. Further Evaluation on FEC Redundancy Estimators

We next study further how each parameter (i.e., k, N', and δ) affects the accuracy of the FEC redundancy estimators. We only investigate the *RS* method, as the *FSLF* approach is equivalent to *RS* when $N' = N$ and yields better accuracy otherwise, as shown in Fig. 2(b).

In this simulation, we focus on the learning phase. The parameters are $M = 1000$, $N = 1000$ and the packet loss rates among receivers are again heterogeneous, and uniformly distributed in the range of $[0, 20\%]$. The estimations are compared with the empirical result which is obtained from Eq. (1) by running 10^5 identical iterations, and each point in the figures represents an average over 100 independent estimations, plotted with a a 95% confidence interval.

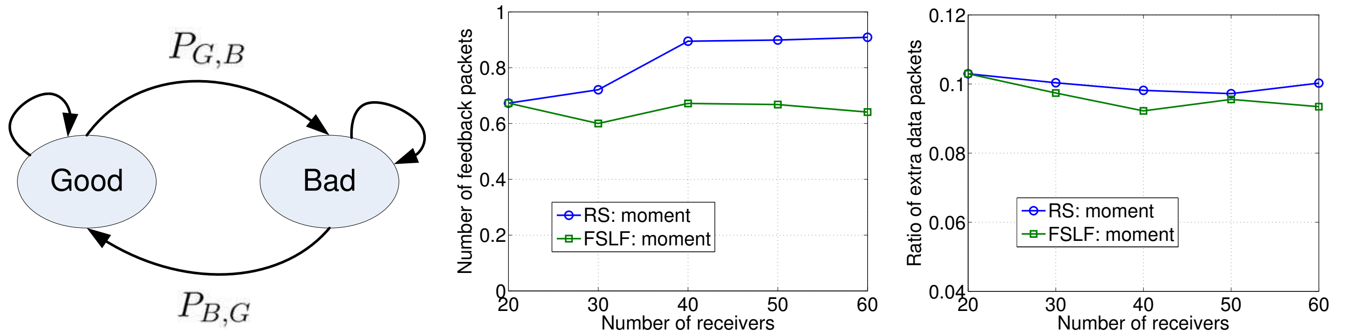
1) Varying the intermediate number k: We first investigate the accuracy of the estimators by varying the choice of the intermediate number k and fixing $N' = N = 1000$, $\delta = 95\%$. Fig. 3(a) shows that the bias of both estimators increases as k increases, which is expected from the properties of extreme quantile estimators [32]. The results show that while a larger value of k leads to a smaller variance for the moment estimator, it yields a larger variance for the maximum likelihood estimator. Although the moment estimator has larger variance for small values of k (e.g., $k = 10, 20$), its 95% confidence interval is still small, i.e., less than 1% of the estimated quantile.

2) Varying the sample size N': We next study how the



(a) (b) (c)

Fig. 3. Performance of extreme value FEC redundancy estimators for different parameters, with 95% confidence interval. (a) $N' = 1000$, $\delta = 95\%$, varying k, (b) $\delta = 95\%$, varying N', (c) $N' = 1000$, varying δ.



(a) (b) (c)

Fig. 4. Extreme value estimators for network with small number of receivers with non-i.i.d. packet loss rates. (a) Two state error-burst channel model, (b) The amount of feedback, (c) The amount of extra data communication.

sample size N', affects the estimators' accuracy, by fixing $\delta = 95\%$, $k = 10, 20$ for the moment estimator, and $k = 50, 150$ for the ML estimator. As expected, Fig. 3(b) shows that the estimators become more accurate and their variances reduce as N' increases. It is worth noting that for the moment estimator, a sample size $N' = 25$ is sufficient to achieve good estimation, i.e., within 4% of the empirical value. This implies that the overhead of using *RS* can be made very small, since it roughly corresponds to N' feedback packets, as shown in Fig. 2(a).

3) Varying the success probability δ: We verify the accuracy of the estimators over a large range of desired success probability δ, ranging from 0.7 to 0.9995. The result (Fig. 3(c)) shows that both estimators are accurate over the entire range. As one could expect, the estimation variance increases with the stringency of the success probability. However, all estimation errors are within the order of 5% of the empirical value.

C. Small Network and Non-i.i.d. Scenarios

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 error-burst channel model in Fig. 4(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 an 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 switches back to the good state with probability $P_{B,G} = 0.5$ at subsequent time slots.

Similar to previous simulations, the source first transmits one page, collects feedback, 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. 4(b) and Fig. 4(c) demonstrate that for these scenarios, the estimators still significantly reduce feedback 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.

Both RS and FSLF can be adapted for time-varying channels by repeatedly running the learning phase periodically, depending on channel coherence times. From an engineering perspective, slightly overestimating the δ-*reliable volume* and sending a few extra encoded packets will effectively ensure that at least δ fraction of the receivers successfully decode the page.

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 Rateless Deluge [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 20 packets, each with 23 bytes of payload. 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 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 feedback packets 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 ($T_{report} = 500$ (ms), as in [1]). In the case that a receiver requests additional encoded packets during this interval, the packets are transmitted. Otherwise the source proceeds to the next page. Note that it is possible that the source will proceed to next page without delivering the current page to all nodes. However, the underlying Deluge protocol, upon which EV-QE Deluge is based, guarantees that all data will be reliably disseminated to all nodes eventually [1].

B. Tmote Sky Sensor Testbed

The performance of EV-QE Deluge and the original Rateless Deluge is first evaluated on a testbed with 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. During each experiment, a new file is injected from a PC into the base station to disseminate to the receivers.

The size of the file is 8518 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 feedback from every receiver after disseminating the first page. Namely, $N' = N = 12$ in the first experiment. The intermediate number k is set to 5.

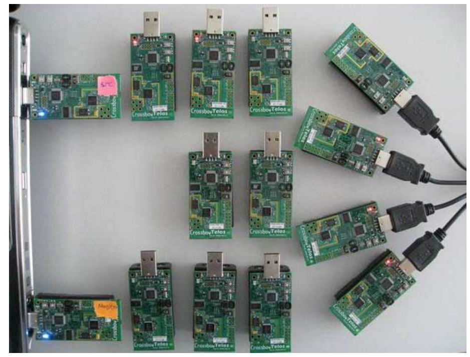


Fig. 5. Tmote Sky sensors testbed with 14 motes.

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

Fig. 6. Rateless Deluge vs. EV-QE Deluge: 20 pages, $N = 12$, $M = 20$, heterogeneous packet loss.

The results in Fig. 6 show that EV-QE Deluge sends out slightly more encoded packets (about 6%). However, it drastically reduces the amount of feedback, which is only 17.3 packets 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. 6. Therefore, with EV-QE Deluge, in the transmission phase, the average number of feedback packets 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 is also monitored through PowerTOSSIM [33]. 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. 7(a), 7(b) and 7(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 feedback packets of EV-QE Deluge remains almost constant at about 50, including the $N' = 30$ initial feedback packets for the source to estimate δ-*reliable volume*. On the other hand,

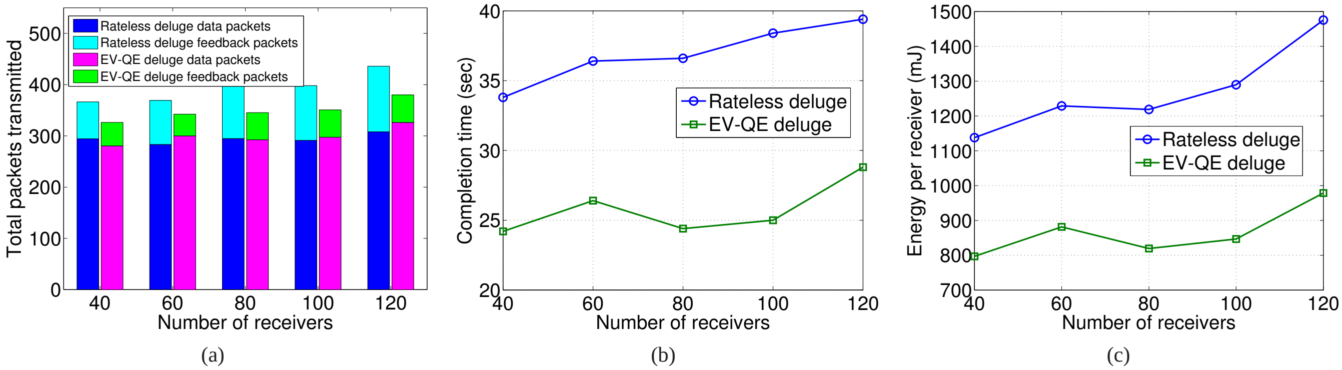


Fig. 7. Rateless Deluge vs. EV-QE Deluge, TOSSIM simulation: 9 pages, $M = 20$, heterogeneous packet loss, varying the number of receivers N. (a) Total packets transmitted: forward and feedback channels, (b) Completion time, (c) Average energy consumption per receiver.

the amount of feedback 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 receiver by about 30%.

VIII. Application to Real Time Oblivious (RT) Rateless Codes

A. RT Codes

To further emphasize the general applicability of our results, we describe in this section application of extreme value estimators to real time (RT) oblivious codes. RT codes are erasure correcting rateless codes which use a feedback channel from the receiver to the source in order to efficiently encode packets at the source. As compared to other rateless codes that require very few redundant packet transmissions, RT codes trade communication efficiency (encoded packets transmitted, feedback) for lower processing overhead and lower memory requirement at the receivers. To achieve this, a receiver discards any encoded packet that cannot be decoded immediately; therefore RT encoded packets are designed to maximize the decoding probability of encoded packets when they are received.

The RT encoder creates each encoded packet by combining (XORing) d randomly-chosen input packets out of the M total input packets ($d \leq M$), where d is the degree of the encoded packet. Let m be the number of input packets already decoded at the receiver and reported to the source (encoder) via feedback. The degree d is determined as follows

$$d = \begin{cases} M & \text{if } m = M-1 \\ \left\lfloor \frac{M+1}{M-m} \right\rfloor & \text{otherwise,} \end{cases} \tag{24}$$

The encoder continues transmitting encoded packets as described above until the receiver has decoded all the input packets *i.e.*, until $m = M$. Using the construction in Eq. (24) the authors in [14] show that the expected number of encodings required for decoding M input packets is less than $2M$. The expected number of feedback messages from the receiver to the source is $O(\sqrt{M})$, and the total expected decoding complexity of RT codes is $O(M \log M)$.

A receiver can decode a degree d encoded degree packet if any $d-1$ input packets used its construction are already available (previously decoded) at the receiver; because XORing the encoded packet with these $d-1$ input packets reveals an unknown input packet. Otherwise the encoded packet is discarded by the receiver (instead of being stored for decoding at a later time, as is the case, for example, in LT [8] decoding). When an input packet is successfully decoded the receiver may elect to send its decoding progress (the updated number of input packets decoded, m) to the source if doing so changes the degree d of encoded packets in Eq. 24.

Consider the scenario of extending this to the case of a source transmitting M input packets to N receivers in a wireless communication environment with i.i.d. packet loss rates across the receivers. The value of m across the receivers may vary significantly during decoding. In order to accommodate all the N receivers, the source has to encode packets using the smallest value of m collected from the receivers. Otherwise, some receivers will not be able to decode the packets.

However, the most significant problem with this approach is that the number of feedback packets from receivers to the source grows as $O(N\sqrt{M})$, as we shall demonstrate in Section VIII-C. For large receiver populations, as is often the case in dense cellular and sensor networks, the source would be overwhelmed by the number of feedback packets.

B. Broadcasting Version of RT code

To improve the applicability of RT codes in a broadcasting scenario, we incorporate use of extreme value estimation techniques. Thus, instead of collecting feedback from the receivers to adjust the RT degree distribution of the encoded symbols[1], we propose to have the source accurately predict these timings. We consider the same problem as in the previous sections, i.e., broadcasting a file with multiple pages from a source to N receivers within its communication range. Each page consists of M packets. Encoding is done at the packet level using an RT code. In the broadcasting version of RT code, the source adjusts the degree of encoded packets according to the number of decoded packets of each receiver using Eq. (24). Specifically, let the number of input packets decoded at receiver n to be

[1]The terms ‘symbol’ and ‘packet’ are used interchangeably in this text.

m_n $(n = 1, .., N)$, then the source creates a degree d encoded packet according to the following equation,

$$d = \begin{cases} M & \text{if } \min_n m_n = M - 1 \\ \lfloor \frac{M+1}{M - \min_n m_n} \rfloor & \text{otherwise.} \end{cases} \quad (25)$$

In our approach, the source collects feedback in the form of sample data from a few receivers and estimates the transition points when the encoded packets' degrees are to be incremented. In effect, the source broadcasts encoded packets and adjusts their degrees according to the total number of encoded packets already broadcast instead of relying on continuous feedback from the receivers.

Note that this problem differs from the previous sections in that here the estimation is performed to predict *multiple* transition points at a time. Specifically, denote by $\theta_{n,m}$ the number of encoded packets the source needs to broadcast for node n to be able to decode m input packets. The source can determine the degree of the encoded packets with information about $\theta_{n,m}$ from all receivers. For example, if according to the original RT code design, the degree of the encoded packet is d when all receivers have decoded m packets, then alternatively, the source can adjust the degree to be d when $\max_n \theta_{n,m}$ packets have been sent. Therefore, the problem becomes to estimate $\max_n \theta_{n,m}$ by only collecting a small amount of feedback when transmitting the first page, instead of continuously collecting feedback from all receivers for each page.

Our goal is to sample and analyze a fixed number of feedback packets in the broadcast of the first page of a file, in order to estimate the *δ-reliable volume* t_δ of each instance in the RT code design when the degree of encoded symbols changes. We can then reduce the amount of feedback while transmitting subsequent pages by having the source broadcast encoded packets according to the estimation instead of feedback from the receivers.

The sampling technique used here is random sampling, i.e., *RS* as described in Section V-A. For simplicity, we only consider the moment estimator. Similar to EV-QE Deluge, our proposed RT code with EVT estimation technique first obtains estimations by transmitting the first page. It then uses these estimations for the transmission of the rest of the pages. If after the transmission of a page, there remain one or more receivers which have not finished receiving it, the source switches back to the original RT code.

C. RT Codes Simulation Results

We evaluate the performance of our EVT-based broadcasting version of RT code, namely, RT$_B$ scheme (labeled EVT estimation in the figures). The number of receivers in the network is $N = 100$. We assume the packet loss rates across the receivers are heterogeneous, unknown, and they are i.i.d and uniformly distributed between 10% and 20%. The sample size of *RS* is 15 and the intermediate number for extreme value estimator is $k = 10$. The success probability associated with each transition point to estimate is $\delta = 99\%$. The simulation results shown here represent an average over 100 independent identical iterations.

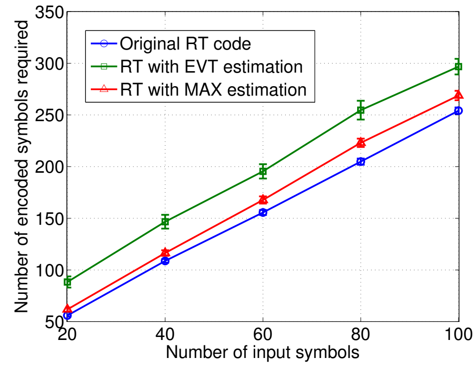


Fig. 8. Expected number of transmissions from the source. $N = 100$ receivers, packet loss rates are distributed uniformly at random from 10% to 20%, varying the number of input packets. 95% confidence interval. Averaged over 100 iterations.

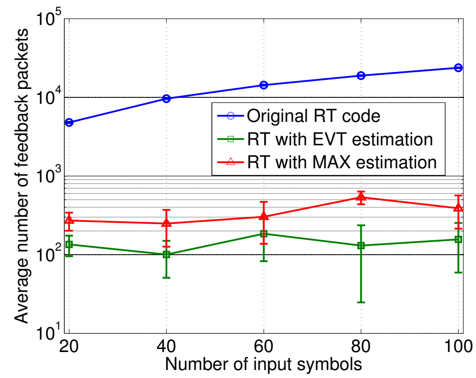


Fig. 9. Expected number of feedback packets while varying the number of input symbols. $M = 100$ input packets, packet loss rates are distributed uniformly at random from 10% to 20%, varying the number of receivers. 95% confidence interval. Averaged over 100 iterations.

In the following simulations we evaluate the performance of the original RT code, RT$_B$ and RT code with another simple estimation technique (labeled MAX estimation). With the MAX estimation, the source determines the degree by simply taking a maximum of the given $N'(N' \leq N)$ sample data, $\theta_{1,m}, .., \theta_{N',m}$, collected during the transmission of the first page. Namely, instead of performing extrapolation using Eq. (5), the MAX estimation simply uses the largest order statistic, i.e., $\max_{1,..,N'} \theta_{n,m}$ to estimate the actual shifting point, $\max_{1,..,N} \theta_{n,m}$. Note that while this approach is simple, it generally underestimates the transition point, since the sample size is much smaller than the number of receivers in the network. Moreover, this simple approach does not provide any relationship between the number of packets broadcasted by the source and the probability of successfully delivering the pages.

Figs. 8 and 9 plot the average number of encoded packets and the average number of feedback packets needed to guarantee completion across all receivers for the original RT code, EVT estimation and MAX estimation. Both estimation techniques need slightly more encoded packets than the original RT codes. The difference in the number of encoded packets required

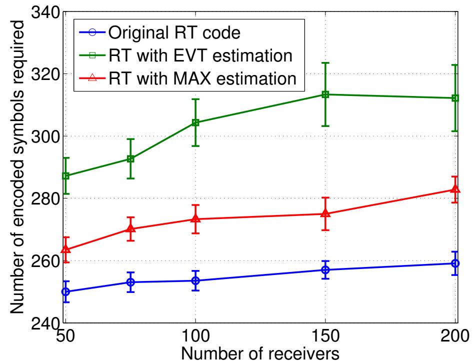


Fig. 10. Original RT vs. modified RT, expected completion time. $M = 100$ input packets, packet loss rates are distributed uniformly at random from 10% to 20%, varying the number of receivers. 95% confidence interval. Averaged over 100 iterations.

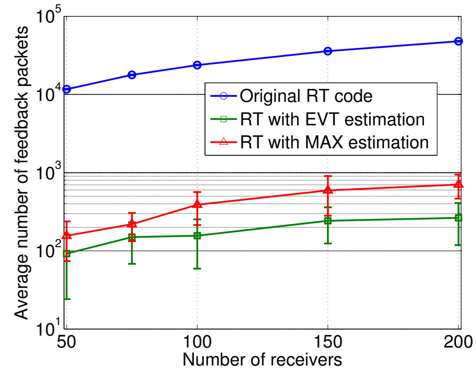


Fig. 11. Expected number of feedback packets while varying the number of receivers. $M = 100$ input packets, packet loss rates are distributed uniformly at random from 10% to 20%, varying the number of receivers. 95% confidence interval. Averaged over 100 iterations.

by EVT estimation and the original RT code remains almost constant even as the number of input packets (M) increases. However, both the EVT and MAX estimation techniques drastically reduce the amount of feedback required as compared to the original RT codes.

MAX estimation transmits less encoded packets than EVT estimation because it underestimates the time to change the degree. This is because MAX estimation may fail to take into account very slow receivers, and may therefore be too optimistic about the decoding rate of the estimated slowest receiver. This results in receivers falling back to the original RT scheme more often with MAX estimation and, consequently, significantly more feedback.

In Figs. 10 and 11, we compare the performance of the different schemes while varying the number of receivers. We report the average number of encoded packets required and the number of feedback packets needed to guarantee completion.

For all sizes of receiver populations, the number of encoded packets transmissions required by the EVT and MAX estimation techniques is slightly larger (10% $\sim$ 20%) than that required by original RT codes, and appears to grow sub-linearly with the number of receivers. On the other hand, the number of feedback packets required using the estimation techniques is drastically smaller (revealing a reduction by a multiplicative factor of 60 with MAX and 150 with EVT) than that required by the original RT codes. Though the size of feedback packets is generally smaller than that of data packets, in many cases the difference is not very significant. For instance, in TinyOS [1], the default maximum packet size (including headers) is 36 bytes while the minimum packet size is 13 bytes.

IX. Concluding Remarks

In this paper, we propose novel, on-line prediction mechanisms for data dissemination in wireless networks with heterogeneous packet loss probabilities. These mechanisms are based on a combination of rateless coding with extreme value theory (EVT) estimation. Rateless coding requires receivers to only receive a sufficient number of distinct, encoded packets and eliminates the need for them to convey control information about *which* specific packets require retransmission (i.e., they only need to indicate the *number* of missing packets). Following a short learning phase, EVT estimation nearly suppresses feedback packets and retransmissions altogether by providing a source with an accurate prediction of the number of redundant packet transmissions needed. 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 amount of feedback 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. We show that *FSLF* using the moment estimator provides the best trade-off in terms of obtaining high estimation accuracy while maintaining low feedback. We then further investigate the impact of the system parameters on the estimation result through simulation and provide guidelines for practical implementation. We also 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.

Finally, we incorporate use of EVT estimation into RT codes under a broadcasting scenario. We employ EVT to estimate the transition points (i.e., the number of packets transmissions), at which a source changes the degree of encoded packets. Our simulations show that such an approach reduces the total number of feedback packets by a factor of 100 compared to original RT codes. These results demonstrate the wide applicability of our protocols to improving the performance of any broadcasting application making use of feedback.

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 and W. Xiao, “Asymptotically Optimal Data Dissemination in Multi-Channel Wireless Sensor Networks: Single Radios Suffice,” *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 695–707, June 2010.

[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, USA: IEEE Computer Society, 2002, p. 271.

[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] A. Beimel, S. Dolev, and N. Singer, “RT oblivious erasure correcting,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1321–1332, 2007.

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

[16] J. Nonnenmacher, E. Biersack, and D. Towsley, “Parity-based loss recovery for reliable multicast transmission,” in *SIGCOMM ’97*, 1997.

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

[18] M. Mosko and J. J. Garcia-Luna-Aceves, “An analysis of packet loss correlation in FEC-enhanced multicast trees,” in *ICNP ’00*, 2000.

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

[20] W. Xiao and D. Starobinski, “Extreme Value FEC for Reliable Broadcasting in Wireless Networks,” *IEEE JSAC Special Issue on Simple Wireless Sensor Networking Solutions*, vol. 28, no. 7, pp. 1180–1189, Sept. 2010.

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

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] L. L. Cam, “Maximum Likelihood: An Introduction,” *International Statistical Review / Revue Internationale de Statistique*, vol. 58, no. 2, pp. 153–171, Aug. 1990.

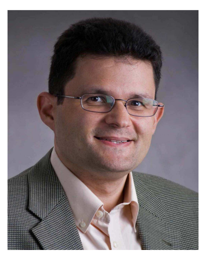
[31] S. D. Grimshaw, “Computing Maximum Likelihood Estimates for the Generalized Pareto Distribution,” *Technometrics*, vol. 35, no. 2, pp. 185–191, 1993.

[32] Frederico Caeiro and M. Ivette Gomes, “Minimum-Variance Reduced-Bias Tail Index and High Quantile Estimation,” *REVSTAT - Statistical Journal*, vol. 6, no. 1, pp. 1–20, March 2008.

[33] 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.

Weiyao Xiao received his B.Eng. from Harbin Institute of Technology, Harbin, China in 2004. He received his M.Sc. and Ph.D. in Electrical Engineering from Boston University, Boston, USA in 2007 and 2010, respectively. He has interned at SAP Labs, LLC, Palo Alto, CA, USA in 2006. He has also interned at Deutsche Telekom A.G., Laboratories (T-Labs) in Berlin, Germany in 2009. His research interests include reliable data dissemination in wireless networks, statistical estimation.

Sachin Agarwal is a senior research scientist at Deutsche Telekom A.G.’s research laboratory (T-Labs) in Berlin, Germany. He received his Ph.D. and M.S. degrees from Boston University’s Networking and Information Systems Laboratory in the department of Electrical and Computer Engineering. He has interned at Microsoft Research in Redmond, USA, in the Communications and Collaboration Systems group and worked for a technology start-up in Boston, USA, in the past. He was a visiting scholar in Stanford University’s Electrical Engineering department in summer 2008.



David Starobinski received his Ph.D. in Electrical Engineering (1999) from the Technion-Israel Institute of Technology. In 1999-2000 he was a visiting post-doctoral researcher in the EECS department at UC Berkeley. In 2007-2008, he was an invited Professor at EPFL (Switzerland). Since September 2000, he has been with Boston University, where he is now an Associate Professor.

Dr. Starobinski received a CAREER award from the U.S. National Science Foundation and an Early Career Principal Investigator (ECPI) award from the U.S. Department of Energy. He is currently an Associate Editor of the IEEE/ACM Transactions on Networking. His research interests are in the modeling and performance evaluation of high-speed, wireless, and sensor networks.

Ari Trachtenberg received his Ph.D. (2000) and MS (1996) in Computer Science from the University of Illinois at Urbana/Champaign, and his S.B. from MIT (1994) in math with CS. He is currently an Associate Professor of Electrical and Computer Engineering at Boston University, where he has been since September 2000.

Dr. Trachtenberg has worked with the Johns Hopkins Center for Talented Youth, Hewlett Packard Laboratories, and MIT Lincoln Lab. He is currently a BU College of Engineering Innovative Engineering Education Faculty Fellow (2009), and has received the ECE award for excellence in teaching (2003), an NSF CAREER (2002), Kuck Outstanding Thesis at UIUC (2000), Mavis Memorial Fund scholarship (1999), and CSE and University fellow (1994). His research interests include applications of error correcting codes to networking, algorithms, and security.