On the Scalability of Reliable Data Transfer in High Speed Networks

Majid Ghaderi† and Don Towsley‡

Abstract

This paper considers reliable data transfer in a high-speed network (HSN) in which the per-connection capacity is very large. We focus on sliding window protocols employing selective repeat for reliable data transfer and study two reliability mechanisms based on ARQ and FEC. The question we ask is which mechanism is more suitable for an HSN in which the scalability of reliable data transfer in terms of receiver’s buffer requirement and achievable delay and throughput is a concern. To efficiently utilize the large bandwidth available to a connection in an HSN, sliding window protocols require a large transmission window. In this regime of large transmission windows, we show that while both mechanisms achieve the same asymptotic throughput in the presence of packet losses, their delay and buffer requirements are different. Specifically, an FEC-based mechanism has delay and receiver’s buffer requirement that are asymptotically smaller than that of an ARQ-based selective repeat mechanism by a factor of $\log W$, where $W$ is the window size of the selective repeat mechanism. This result is then used to investigate the implications of each reliability mechanism on protocol design in an HSN in terms of throughput, delay, buffer requirement, and control overhead.

Index Terms

Reliable data transfer, scalability, high-speed networks, selective repeat.

†Department of Computer Science, University of Calgary, Email: mghaderi@ucalgary.ca
‡Department of Computer Science, University of Massachusetts Amherst, Email: towsley@cs.umass.edu
On the Scalability of Reliable Data Transfer in High Speed Networks

I. INTRODUCTION

The role of a reliable data transfer protocol is to deliver data from a traffic source to a receiver such that no data packet is lost or duplicated, and all packets are delivered to the receiving application in the order in which they were sent. Currently, TCP is the dominant protocol for reliable transmission of data in the Internet. The reliability mechanism in TCP is based on a sliding window protocol in which the set of packets that can be transmitted by the sender at any time instant is determined by a logical window over the stream of incoming packets. The size of the window is governed by the TCP congestion control algorithm known as additive increase multiplicative decrease (AIMD). The poor performance of AIMD in a high-speed network (HSN) has been subject to numerous studies, and consequently, several modifications have been proposed to remedy its shortcomings [1].

In this work, we argue that another fundamental problem of TCP in an HSN is its reliability mechanism. This problem is particularly manifested in an HSN where there is a non-zero probability of packet loss, for example, due to small router buffers [2]. It has been shown that high-speed TCP variants achieve poor bandwidth utilization due to small buffers unless the loss probability at routers is maintained at non-negligible levels, e.g., 1% (see [3]). This results in a large number of lost packets that need to be retransmitted by TCP, causing out-of-order packets, and consequently long delays and a large buffer requirement at the receiver.

We consider two sliding window reliable data transfer protocols based on Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). A sliding window protocol allows the sender to have multiple on-the-fly packets, which have been transmitted but not acknowledged yet. Typically, the channel between the sender and receiver is lossy, and hence some packets may not reach the receiver. To implement reliability, the receiver sends feedback to the sender over a feedback channel. In this work, we assume that the sender and receiver implement the selective

---

1In this paper, we only consider unicast protocols.
repeat protocol (SR) for reliable transmission of packets. In the selective repeat protocol, the sender can transmit multiple packets before waiting for feedback from the receiver, while the receiver accepts every packet that arrives at the receiver.

There are several important performance measures associated with the selective repeat protocol: throughput, buffer occupancy at the sender and receiver, packet delay, and protocol overhead. For reliable data transfer, at the receiver, packets have to be delivered to the application in-order. Packets can arrive at the receiver out-of-order, for example, some packets may be lost causing gaps in the sequence of received packets. Thus, the receiver may need to buffer the arriving packets for some time until they can be delivered in-order. An important metric in analyzing the selective repeat protocol is the so-called re-sequencing delay at the receiver’s buffer. The re-sequencing delay is the delay between the time when a packet is received at the receiver and the time it is being delivered to the application. The set of packets in the receiver’s buffer is called the re-sequencing buffer occupancy. The goal of this paper is to analyze the receiver’s buffer occupancy, packet delay and protocol overhead of the selective repeat protocol based on ARQ and FEC reliability mechanisms in a setting where both protocols achieve their respective maximum throughput.

Traditionally, SR is coupled with ARQ to recover from packet losses. In this paper, in addition to the traditional SR, we consider another variation of the protocol in which ARQ is augmented with FEC at the sender to enable the receiver to recover from packet losses more efficiently. The intuition is that, with FEC, packets do not have individual identities, and hence as long as a sufficient number of them arrive at the receiver, the receiver is able to decode and deliver packets in-order to the application. In contrast, with pure ARQ, transmitted packets have unique identities, and hence the receiver has to receive at least one copy of every packet before it can deliver the packets in-order to the application. As shown in this paper, it takes a longer time and requires a larger buffer space for the receiver to collect a copy of every packet as opposed to just collecting a given number of packets.

Throughout the paper, we use the term “high-speed network” to refer to a network where the per-connection capacity approaches infinity regardless of the number of connections in the network. We show that in an HSN, reliability mechanisms based on FEC are more suitable as their buffer and delay performance is superior to that of ARQ based reliability mechanisms, while asymptotically achieving the same throughput. Specifically, we show that the ARQ and
FEC based mechanisms, respectively, require $\Theta(W \log W)$ and $\Theta(W)$ buffer space, and achieve $\Theta(\log W)$ and $\Theta(1)$ delay, where $W$ denotes the window size in the selective repeat protocol.

We note that employing FEC based reliability mechanisms in TCP has been proposed in the literature. LT-TCP was proposed in [5] to augment TCP with adaptive FEC to cope with high packet loss rates in wireless networks. TCP/NC proposed in [6] incorporates network coding as a middleware below TCP to deal with packet losses. Our focus in this paper is to characterize the asymptotic performance of ARQ and FEC based mechanisms in HSNs to determine which mechanism is more scalable when per-connection capacity is very large.

Our contributions can be summarized as:

• We derive closed-form expressions for the performance of ARQ and FEC based reliability mechanisms in terms of throughout, receiver’s buffer and delay for an arbitrary window size $W$.

• We characterize the asymptotic performance of the ARQ and FEC based reliability mechanisms as the per-connection capacity, and consequently the window size $W$, grows to infinity.

• The asymptotic results are used to compare the performances of the two mechanisms in terms of throughput, delay, buffer requirement and protocol overhead.

The rest of this paper is organized as follows. In Section II, we describe the reliability mechanisms considered in the paper. Sections III and IV are dedicated, respectively, to the exact and asymptotic analysis of ARQ and FEC based reliability mechanisms. Several extensions and implications of our results on protocol design are discussed in Section V. Section VI reviews some related work, while our concluding remarks are presented in Section VII.

II. RELIABLE DATA TRANSFER PROTOCOLS

In the following subsections, we describe the operation of two selective repeat protocols based on ARQ and FEC for reliable data transfer in lossy networks. Our objective is to study the asymptotic performance of these protocols, and hence we focus on the basic operation of each protocol and ignore complications that arise due to variable round-trip-times, channel reordering and imperfect feedback.

In traditional SR, the window size $W$ is used to limit the buffer space required at the sender and receiver by limiting the difference between the greatest and smallest sequence numbers of
outstanding packets (packets that have been transmitted but not acknowledged yet) to $W$. Such a mechanism requires exactly $W$ buffer space at the sender and receiver but has a throughput penalty, and thus is unable to fully utilize the available network capacity. For instance, if every packet in a window is acknowledged except the first packet, the sender is not allowed to transmit any new packets since the window is exhausted. In contrast, we allow our protocols to have up to $W$ outstanding packets regardless of the range of the sequence numbers of the transmitted packets. Clearly, this approach allows SR to fully utilize the available capacity, which is desirable in a HSN, but requires larger buffer space at the sender and receiver. Our objective in this paper is to characterize the increased buffer requirement and corresponding packet delay at the receiver.

A. ARQ Based Selective Repeat Protocol

The ARQ-based protocol, referred to as SR-ARQ, employs ARQ for reliable data transmission. The sender continuously transmits new packets in increasing order of sequence numbers as long as ACKs are received for the transmitted packets. For every packet arriving at the receiver, the receiver sends an ACK to the sender over the feedback channel. The ACK packet contains the sequence number of the packet for which the ACK is being transmitted. After transmitting a packet, the sender transmits up to $W - 1$ subsequent packets (new or retransmitted). If an ACK arrives at the sender during this period, the corresponding packet is removed from the sender buffer and a new packet is subsequently transmitted. If time-out occurs, i.e., no ACK arrives during this time, then the same packet is subsequently transmitted again. The receiver buffers out-of-order packets for later in-order delivery to the application. A packet is delivered to the application and removed from the receiver’s buffer only when all packets with smaller sequence numbers have been released from the buffer.

Algorithm 1 summarizes the main events and their corresponding actions at the sender and receiver under SR-ARQ.

B. FEC Based Selective Repeat Protocol

The FEC-based protocol, referred to as SR-FEC, is a selective repeat protocol that implements a rateless coding algorithm for packet transmission. At the sender, packets are divided into consecutive coding blocks for transmission to the receiver. Each coding block contains $B > 1$ packets, which are coded together to form coded packets. A consecutive sequence number
Algorithm 1 SR-ARQ events and actions.

**Sender:** \{\(s^*\) : next packet sequence number\}

- **ACK**\((s)\): \{arrival of ACK for packet \(s\)\}
  1) mark packet \(s\) as ACKed
  2) remove in-order ACKed packets from buffer
  3) transmit packet \(s^*\) and start a timer for it
  4) \(s^* \leftarrow s^* + 1\)

- **Timeout**\((s)\): \{time-out for packet \(s\)\}
  1) retransmit packet \(s\) and start a timer for it

**Receiver:**

- **Receive**\((s)\): \{arrival of packet \(s\)\}
  1) transmit ACK \(s\)
  2) copy packet \(s\) to buffer
  3) remove in-order packets from buffer and deliver them to application

is assigned to each block to uniquely identify coding blocks. Every coded packet carries its corresponding block sequence number in its header in addition to a packet sequence number that is unique within its corresponding block. Thus, the pair of block and packet sequence numbers uniquely identifies a packet.

While SR-FEC considered in this paper is independent of the specific coding technique used to generate coded packets, for the sake of concreteness, we assume that the sender performs random fountain coding \[7\] to generate coded packets. In random fountain coding, at every transmission instance, a subset of the packets in the current coding block is chosen randomly and the packets in the subset are XOR’ed together to create a coded packet. The coded packet is then transmitted to the receiver. Random fountain codes are a special class of fountain codes where the degree distribution of coded packets is uniform. While their decoding complexity is higher than LT and Raptor codes, they exhibit similar error correction performance, which is the reason for their consideration in this paper.

The receiver decodes packets in a block-by-block manner. To decode a block, \(B\) independent coded packets from that block are required. Upon receiving a coded packet, the receiver infers
which original packets were XOR’ed to create the newly received coded packet. When the block+packet sequence numbers are unique, the coding and decoding algorithms at the sender and receiver can be synchronized, hence avoiding the need to include any extra information in packet headers to help infer the identity of the XOR’ed packets. This can be implemented, for example, by synchronizing a random number generator at the receiver with the random number generator used at the sender for choosing the set of XOR’ed packets.

Similar to SR-ARQ, the sender continuously transmits coded packets in increasing order of coding block sequence number as long as ACKs are received. For every innovative coded packet (i.e., a packet that is independent of the previously received coded packets for the corresponding block) arriving at the receiver, the receiver sends an ACK to the sender over the feedback channel. The ACK only contains the sequence number of the coding block for which the ACK is being transmitted. After transmitting a packet, the sender transmits up to \( W - 1 \) subsequent coded packets. The arrival of an ACK arrives at the sender during this period triggers a new coded packet. The new packet is constructed from the coding block that contains the packet. If a time-out occurs, i.e., no ACK arrives, then a coded packet from the coding block that caused the time-out is subsequently transmitted. When \( B \) ACKs are received for a coding block then it is released from the sender window.

Algorithm 2 summarizes the main events and their corresponding actions at the sender and receiver under SR-FEC.

### III. Exact Performance Analysis

An important metric in characterizing the performance of ARQ protocols is the expected number of retransmissions required at the sender until a packet arrives successfully at the receiver. Our focus in this section is on deriving exact expressions for this metric. Using this metric, we show how it can be used to compute the receiver’s buffer occupancy and packet delay for SR-ARQ and SR-FEC.

#### A. Assumptions

We consider a heavy traffic situation, in which packets arrive at the sender from an infinite source. For any arrival process, this model provides upper bounds on the receiver buffer requirements and packet delay. It is assumed that all packets are of the same size, and that the
Algorithm 2 SR-FEC events and actions.

**Sender:** \( \{b^* : \text{next block sequence number}\} \)

- **ACK(\(b\)):** \( \{\text{ACK for block } b\} \)
  1) if block \(b\) has \(B\) ACKs then mark it as ACKed
  2) remove in-order ACKed blocks from buffer
  3) transmit a coded packet for block \(b^*\) and
     start a timer for it
  4) if \(b^*\) has \(B\) pending and received ACKs then
     \(b^* \leftarrow b^* + 1\)

- **Timeout(\(b\)):** \( \{\text{time-out for block } b\} \)
  1) retransmit a coded packet for block \(b\) and start a timer for it

**Receiver:**

- **Receive(\(b\)):** \( \{\text{arrival of coded packet for block } b\} \)
  1) transmit ACK \(b\)
  2) copy the packet to receiver buffer for block \(b\)
  3) if block \(b\) is full rank then decode block \(b\)
  4) remove in-order blocks from buffer and deliver them to application

Packet loss process is independent of the packet size. This represents a scenario where packet losses are primarily due to congestion at routers along the path between the sender and receiver. Specifically, we assume that packet losses form a Bernoulli process with mean \(p\).

The analysis presented in this paper assumes that the sender window size is fixed. We note that in TCP, the sender window size is dynamic and changes in response to network congestion situation and available buffer at the receiver. In such cases, our model is not accurate and should be extended to dynamic window sizes. However, in situations where TCP congestion window is stable for long periods of time, our results can be applied. For example, networks with low loss rate (which are of interest in this work), if the TCP congestion window is limited by the receiver’s advertised window, then the sender window size remains constant for long periods of time. In a sense, our model assumes decoupled reliability from congestion control, as advocated for in [3], [8], [9].
To simplify the analysis, as commonly assumed in the literature [10]–[13], we assume that a reliable feedback channel exists between the sender and receiver. The receiver sends feedback in the form of acknowledgement packets (ACKs) to the sender, which are assumed to be delivered over the feedback channel reliably. We also assume that the round-trip-time between the sender and receiver is fixed and equal to $R$.

B. Performance Metrics

In high-speed networks, to efficiently utilize network bandwidth, the window size $W$ should grow proportional to the per-connection capacity. This is simply stated as $W = R \cdot C$, where $C$ is the per-connection capacity. Thus, as per-connection capacity increases, the window size should increase as well resulting in increased buffer occupancy at the sender and receiver. We assume that the sender and receiver have sufficient buffer space to allow each protocol, namely SR-ARQ and SR-FEC, to operate efficiently.

With SR-ARQ, every packet received at the receiver is a useful packet, and hence the throughput of SR-ARQ is given by $\rho = (1 - p) C$. However, with SR-FEC, only independent coded packets are useful for the receiver to decode and recover the original packets. To avoid transmitting coded packets that are not independent from the previously transmitted packets, and hence wasting bandwidth, for every active coding block, the sender memorizes the coded packets transmitted so far for that coding block, for example, by keeping track of the indices of the packet subsets used to construct the coded packets. For a given coding block, the sender only sends packets that are independent of the coded packets transmitted previously. In this case, the throughput of SR-FEC is also given by $\rho = (1 - p) C$. The analysis presented in Sections III and IV considers this case. Later, in Section V we address the case when coded packets may be dependent.

Let $Q$ and $D$ denote the re-sequencing buffer occupancy and delay at the receiver, respectively. Using the Little’s law, it follows that

$$\mathbb{E}[D] = \frac{\mathbb{E}[Q]}{(1 - p)C} = \frac{R}{(1 - p)W} \mathbb{E}[Q]. \quad (1)$$

In the rest of the paper, we focus on computing $\mathbb{E}[Q]$ for both protocols.
C. SR-ARQ Analysis

Consider a window of $W$ packets. Let $N_i$ denote the number of times that the packet at position $i$, for $1 \leq i \leq W$, is transmitted until it is received successfully at the receiver. It follows that $N_i$ has a geometric distribution. In other words, the probability that the packet at position $i$ is received successfully in $n$-th transmission is given by

$$P\{N_i = n\} = (1 - p)p^{n-1}, \quad n \geq 1.$$  \hfill (2)

Among the $W$ packets in the window, let $\ell$ denote the last packet that is successfully received at the receiver, i.e., the packet that takes the most number of retransmissions. Let $N_{ARQ}$ denote the number of transmissions required until $\ell$ is received at the receiver. It follows that

$$N_{ARQ} = \max_{1 \leq i \leq W} N_i.$$  \hfill (3)

Since packet losses are assumed to be independent, the following relation holds for the probability distribution of the random variable $N_{ARQ}$,

$$P\{N_{ARQ} \leq n\} = (1 - p^n)^W, \quad n \geq 1.$$  \hfill (4)

Using the above expression, it follows that,

$$\mathbb{E}[N_{ARQ}] = \sum_{n=1}^{\infty} P\{N \geq n\} = \sum_{n=1}^{\infty} 1 - (1 - p^{n-1})^W.$$  \hfill (5)

Define $f(K)$ for $k \geq 1$ as,

$$f(K) = \sum_{n=1}^{K} 1 - (1 - p^{n-1})^W,$$  \hfill (6)

which, then yields,

$$\mathbb{E}[N_{ARQ}] = \lim_{K \to \infty} f(K).$$  \hfill (7)

Next, we focus on computing a closed-form expression for $f(K)$ that can be used to compute $\mathbb{E}[N_{ARQ}]$:

$$f(K) = K - \sum_{n=1}^{K} (1 - p^{n-1})^W$$

$$= K - \sum_{n=1}^{K} \sum_{i=0}^{W} \binom{W}{i} (-p^{n-1})^i$$

$$= -\sum_{i=1}^{W} \binom{W}{i} (-1)^i \frac{1 - (p^i)^K}{1 - p^i}.$$  \hfill (8)
Taking the limit of $f(K)$ as $K \to \infty$ yields the following result for $\mathbb{E}[\text{N}_{\text{ARQ}}]$:

$$
\mathbb{E}[\text{N}_{\text{ARQ}}] = \lim_{K \to \infty} f(K)
= - \sum_{i=1}^{W} \binom{W}{i} \frac{(-1)^i}{1 - p^i}.
$$

(9)

The above expressions are valid when $0 \leq p < 1$. If $p = 1$, then no packet is received at the receiver and hence the receiver’s buffer will be empty. Over the range $0 \leq p < 1$, as $p$ increases so does the number of lost packets. As the number of lost packets increases, the number of out-of-order packets in the receiver’s buffer increases as well.

D. SR-FEC Analysis

With coding, packets are delivered to the application in a block-by-block manner. For every coding block, the receiver needs to receive $B$ independent coded packets in order to decode and recover the original coding block. Let $N_i$ denote the number of transmitted coded packets until the receiver decodes coding block $i$. The probability that the receiver receives exactly $B$ coded packets after the $n$-th packet transmission by the sender for coding block $i$ is given by a negative binomial distribution, i.e.,

$$
\mathbb{P}\{N_i = n\} = \binom{n-1}{i-1} (1-p)^B p^{n-B}, \quad n \geq B
$$

(10)

which is the probability that $B-1$ packets out of the first $n-1$ transmissions are received successfully and the last transmission is a success too. Recall that, for the moment, we are assuming all coded packets are independent following the mechanism described earlier. Therefore, we obtain that,

$$
\mathbb{P}\{N_i \leq n\} = \sum_{i=B}^{n} \binom{i-1}{B-1} (1-p)^B p^{i-B},
= (1-p)^B \sum_{i=0}^{n-B} \binom{B+i-1}{B-1} p^i.
$$

(11)

Consider a typical window of $W$ packets consisting of $M = W/B$ coding blocks. Without loss of generality, we assume that $W$ is an integer multiple of $B$. Among the $M$ coding blocks, let $\ell$ denote the last block that is decoded by the receiver, i.e., the block that takes the most
number of transmissions until its corresponding $B$ coded packets are received. Let $N_{\text{FEC}}$ denote the number of transmissions required until block $\ell$ is decoded by the receiver. It is obtained that

$$N_{\text{FEC}} = \max_{1 \leq i \leq W} N_i.$$  

(12)

Since packet losses are assumed to be independent, the following relation holds for the probability distribution of the random variable $N_{\text{FEC}},$

$$\mathbb{P}\left\{ N_{\text{FEC}} \leq n \right\} = (1 - p)^W \left( \sum_{i=0}^{n-B} \binom{B+i-1}{B-1} p^i \right)^M.$$  

(13)

However, it seems unlikely that this would result in any simple expression for $\mathbb{E}[N_{\text{FEC}}].$ Note that $N_{\text{FEC}}$ is the number of transmissions per coding block of size $B.$ Thus, the expected number of transmissions per packet is given by $\frac{1}{B} \mathbb{E}[N_{\text{FEC}}].$

IV. ASYMPTOTIC PERFORMANCE ANALYSIS

The exact expressions derived in the previous section do not provide insight about the scaling behavior of the number of transmissions with respect to $W.$ In this section, we derive asymptotic expressions for the expected number of transmissions for both SR-ARQ and SR-FEC. Then, using our asymptotic results, we compute the expected receiver’s buffer occupancy. In particular, we are interested in the asymptotic performance of the protocols as the window size becomes very large, i.e., $W \rightarrow \infty.$ For SR-FEC, we consider two scaling regimes: (1) the coding block length remains constant, i.e., $B \in \Theta(1),$ and (2) the coding block length scales at the same rate as the window size, i.e., $B \in \Theta(W).$

The asymptotic analysis presented in this section utilizes techniques from the extreme value theory [14], which is briefly described next.

A. Extreme Value Theory

Consider a sequence of IID random variables $X_1, X_2, \ldots, X_n$ with a common cumulative distribution function $F,$ i.e., $F(x) = \mathbb{P}\{X_i \leq x\}.$ Let the $X_i$’s be arranged in increasing order of magnitude and denoted by $X_{(1)}, X_{(2)}, \ldots, X_{(n)}.$ These ordered values of $X_i$’s are called order statistics, $X_{(k)}$ being the $k$th order statistics. Note that $X_{(1)}$ and $X_{(n)}$ are the minimum and maximum and are called the extreme order statistics.
Of particular interest in this paper is the asymptotic behavior of \( X(n) \) for large \( n \). Since \( X_i \)'s are independent, the cumulative distribution function of \( X(n) \) is given by

\[
P \{ X(n) \leq x \} = P \{ X(1) \leq x, \ldots, X(n) \leq x \} = F(x)^n,
\]

and, thus, for \( F(x) < 1 \), as \( n \to \infty \),

\[
P \{ X(n) \leq x \} \to 0.
\]

While it is possible to study the exact distribution of \( X(n) \), in many applications, only the behavior of the tail of the distribution is of interest, which is generally difficult to characterize using this exact form. Extreme value theory provides a framework to study the limiting distribution of extreme order statistics.

**Theorem 1** (Extreme Value Theorem [14]). *If there exists a sequence of real constants \( a_n > 0 \) and \( b_n \), such that, as \( n \to \infty \),

\[
P \left\{ \frac{X(n) - b_n}{a_n} \leq x \right\} = F(a_n x + b_n)^n \to G(x),
\]

(14)

for some non-degenerate distribution function \( G \), then the limit distribution \( G \) must be one of the three extreme value distributions as follows.

**Gumbel**: \( G(x) = \exp(-e^{-x}), \quad -\infty < x < \infty \);

**Fréchet**: \( G(x) = \begin{cases} 0, & x \leq 0, \\ \exp(-x^{-\alpha}), & x > 0, \alpha > 0; \end{cases} \)

**Weibull**: \( G(x) = \begin{cases} \exp(-(x)^\alpha), & x \leq 0, \alpha > 0, \\ 1, & x > 0. \end{cases} \)

**Definition 1** (Maximum Domain of Attraction [15]). The maximum domain of attraction of a distribution \( G \) is the set of all distributions \( F \) that has \( G \) as the limiting distribution. That is, as \( n \to \infty \), constants \( a_n > 0 \) and \( b_n \) exist such that (14) holds.

Define symbol \( x^* \) as the upper bound of \( X_1 \), that is,

\[
x^* = \sup \{ x | F(x) < 1 \}.
\]
Definition 2 (von Mises Function [15]). Suppose $F(x)$ is a distribution function with density $f(x)$ which is positive and differentiable on a left neighborhood of $x^*$. If
\[
\lim_{x \to x^*} \frac{d}{dx} \left( \frac{1 - F(x)}{f(x)} \right) = 0, \tag{15}
\]
then $F$ is called a von Mises function.

If distribution function $F$ is a von Mises function then it belongs to the domain of attraction of the Gumbel distribution with a possible choice of the sequence of the normalizing constants $a_n$ and $b_n$ given by
\[
b_n = F^{-1}(1 - \frac{1}{n}), \quad a_n = \frac{1 - F(b_n)}{f(b_n)}, \tag{16}
\]
Noting that for the standard Gumbel distribution, the mean is given by $\gamma$, where $\gamma \approx 0.57721$ is the Euler's constant, it is obtained that, as $n \to \infty$,
\[
\mathbb{E}[X(n)] = a_n \gamma + b_n. \tag{17}
\]

B. SR-ARQ Analysis

Recall that $N_i$, for $1 \leq i \leq W$, has a geometric distribution. Thus, $N_{ARQ}$ is the maximum of $W$ independent and geometrically distributed random variables with the mean $1/(1 - p)$. We will show that $\mathbb{E}[N_{ARQ}] \in \Theta(\log W)$.

Theorem 2. $\mathbb{E}[N_{ARQ}] \in \Theta(\log W)$.

Proof: We approximate each geometric random variable $N_i$ by an exponentially distributed random variable $X_i$ with an appropriate rate $\lambda$. To obtain the same numerical values for the probability distribution of $N_i$ and $X_i$, the following relation must be satisfied:
\[
\mathbb{P}\{N_i \leq n\} = \mathbb{P}\{X_i \leq n\}, \tag{18}
\]
which yields $\lambda = -\ln p$. Define $X$ as follows,
\[
X = \max_{1 \leq i \leq W} X_i. \tag{19}
\]
Next, we show that the approximation error introduced as the result of approximating $\mathbb{E}[N_{ARQ}]$ by $\mathbb{E}[X]$ is bounded by a constant, i.e., $\mathbb{E}[N_{ARQ}] = \mathbb{E}[X] + \Theta(1)$. To compute the approximation error, denoted by $\epsilon$, we note that
\[
\mathbb{E}[X_i] = \sum_{n=0}^{\infty} \int_{n}^{n+1} \mathbb{P}\{X_i > x\} \, dx. \]
Therefore,
\[
\epsilon = \mathbb{E} [N_i] - \mathbb{E} [X_i] = \sum_{n=0}^{\infty} p^n + \frac{1}{\lambda} \sum_{n=0}^{\infty} (e^{-(n+1)\lambda} - e^{-n\lambda})
\]
(20)
\[
= \frac{1}{1-p} + \frac{1}{\ln p}, \quad 0 \leq p \leq 1.
\]
Thus, the problem is reduced to computing the maximum of \( W \) IID exponentially distributed random variables with rate \( \lambda = -\ln p \).

**Lemma 1.** The distribution function of an exponentially distributed random variable is a von Mises function.

*Proof:* See [15]. □

Therefore, there exist constants \( a_W > 0 \) and \( b_W \) so that \((X - b_W)/a_W\) converges in distribution to the standard Gumbel distribution, as \( W \to \infty \). A possible choice for \( a_W \) and \( b_W \) is given by (16), which results in (e.g., see [15]),
\[
a_W = \lambda^{-1}, \quad b_W = \lambda^{-1}(\ln W).
\]
Substituting the above values in (17) yields the following expression for \( \mathbb{E} [X] \),
\[
\mathbb{E} [X] = \frac{\ln W + \gamma}{-\ln p},
\]
(21)
as \( W \to \infty \). Given that \( \mathbb{E} [N_{ARQ}] \) is within a constant of \( \mathbb{E} [X] \), the theorem follows. □

Let \( \mathbb{E} [Q_{ARQ}] \) denote the expected buffer occupancy at the receiver under SR-ARQ. In the following, we establish upper and lower bounds on \( \mathbb{E} [Q_{ARQ}] \), and show that \( \mathbb{E} [Q_{ARQ}] \) scales as \( W \log W \).

**Lemma 2.** \( \mathbb{E} [Q_{ARQ}] \in O(W \log W) \).

*Proof:* Let \( \ell \) denote the packet that requires the most number of retransmissions in a window. In the worst-case, \( \ell \) is the first packet in the window, i.e., the packet with the smallest sequence number, and all packets with sequence numbers greater than that of \( \ell \) are always received correctly at the receiver. In this case, between every retransmission of \( \ell \), \( W - 1 \) new packets with greater sequence numbers are transmitted. The receiver has to buffer all these new packets until the last retransmission of \( \ell \) arrives. As the result, we obtain that
\[
Q_{ARQ} \leq (W - 1) \cdot N_{ARQ},
\]
(22)
and, therefore,

$$\mathbb{E}[Q_{ARQ}] \leq (W - 1) \cdot \mathbb{E}[N_{ARQ}].$$  \hfill (23)

It then follows that $\mathbb{E}[Q_{ARQ}] \in O(W \log W)$.

Lemma 3. $\mathbb{E}[Q_{ARQ}] \in \Omega(W \log W)$.

Proof: To compute a lower bound, consider the case when $\ell$ is the last packet in the window, \textit{i.e.}, has the largest sequence number in the window. As time progresses, some of the packets with smaller sequence number are received at the receiver in-order and consequently delivered to the application. Every new packet that arrives at the receiver with sequence number greater than that of $\ell$ will be placed in the buffer. In the best-case, packets arrive at the receiver in-order so that every time a packet is received it can be delivered to the application and its corresponding buffer is released.

Next, consider the time (in terms of the number of packet transmissions) it takes for the first $W/2$ packets in the window to be received and delivered to the application, assuming that all these packets arrive before packets in the second half of the window (as we are considering the best-case). We show that this time, denoted by $\tau$, is upper bounded by a constant independent of the window size $W$. Thus, after some constant initial time $\tau$, for every retransmission of $\ell$, at least $W/2$ new packets with higher sequence numbers arrive at the receiver that need to be buffered.

We are interested in the expected number of transmissions required to receive the first $W/2$ packets in the best-case. For $n$ IID exponentially distributed random variables with the rate $\lambda$, it is well-known that the time until the first occurrence is given by $1/(n\lambda)$. Thus, the expected time until $W/2$ packets arrive at the receiver is given by,

$$\mathbb{E}[\tau] = \frac{1}{\lambda} \left( \frac{1}{W} + \cdots + \frac{1}{W/2} \right) \leq \frac{1}{\lambda}. \hfill (24)$$

Putting it together, we have,

$$Q_{ARQ} \geq \frac{W}{2} \cdot (N_{ARQ} - \tau), \hfill (25)$$

and, therefore,

$$\mathbb{E}[Q_{ARQ}] \geq \frac{W}{2} \cdot (\mathbb{E}[N_{ARQ}] - \frac{1}{\lambda}). \hfill (26)$$

It then follows that $\mathbb{E}[Q_{ARQ}] \in \Omega(W \log W)$.\hfill \blacksquare
Theorem 3. $\mathbb{E}[Q_{ARQ}] \in \Theta(W \log W)$.

Proof: It follows from Lemmas 2 and 3.

C. SR-FEC Analysis

It is well-known that the performance of rateless coding schemes depends on the size of the coding block. To capture this dependency, we consider two scaling regimes as follows:

I. The size of the coding block remains constant as $W$ grows, i.e., $B \in \Theta(1)$. In this case, the number of coding blocks $M$ grows at the same rate as $W$.

II. The size of the coding blocks grows at the same rate as $W$. In this case, the number of coding blocks remains constant, i.e., $M \in \Theta(1)$, while the size of the coding blocks grows to infinity.

In the rest of this section, we analyze the performance of SR-FEC in these two scaling regimes. We show that, the asymptotic receiver’s buffer occupancy grows by a factor of $\log W$ slower in the second scaling regime compared to the first scaling regime.

Scaling Regime I: Constant Coding Block Size

We show that, in this case, the receiver’s buffer occupancy has the same scaling as with SR-ARQ.

Theorem 4. $\mathbb{E}[N_{FEC}] \in \Theta(\log W)$.

Proof: Recall that the number of transmissions $N_i$ for coding block $i$ has a negative binomial distribution. In an alternative but equivalent representation, $N_i$ can be considered as the sum of $B$ IID geometric random variables with the mean $1/(1-p)$. Each geometric random variable represents the number of transmissions at the sender until the next required coded packet arrives at the receiver. That is,

$$N_i = N_{i1} + \cdots + N_{iB}, \quad (27)$$

where, $N_{ik}$ is the number of transmissions of packets for coding block $i$ at the sender until the receiver receives the $k$th coded packet, given that it has already received $k-1$ coded packets. For geometric random variables we have

$$\mathbb{P}\{N_{ik} = n\} = (1-p)p^{i-n}. \quad (28)$$
Following the approximation method described in the previous section, we approximate each geometric random variable $N_{ik}$ with an exponentially distributed random variable $X_{ik}$ with the rate $\lambda = -\ln p$. Define $X_i$ and $X$ as follows:

$$X_i = X_{i1} + \cdots + X_{iB},$$

and, for $M$ coding blocks in a window,

$$X = \max_{1 \leq i \leq M} X_i. \tag{30}$$

Our goal is to approximate $\mathbb{E}[N_{FEC}]$ with $\mathbb{E}[X]$. From the previous section, the approximation error is given by

$$\epsilon = B \cdot \left( \frac{1}{1 - p} + \frac{1}{\ln p} \right) \in \Theta(B), \tag{31}$$

which is constant.

Next, we turn our attention to computing $\mathbb{E}[X]$. Notice that $X_i$ is the summation of $B$ IID exponentially distributed random variables. Thus, $X_i$ is Erlang distributed with the following cumulative distribution function:

$$F(x) = \mathbb{P}\{X_i \leq x\} = 1 - e^{-\lambda x} \sum_{k=0}^{B-1} \frac{B^{k-1}}{k!}.$$

**Lemma 4.** The distribution function of the sum of $B$ IID exponentially distributed random variables is a von Mises function.

**Proof:** We need to show that condition (15) holds for the Erlang distribution. Equivalently, the following condition should be satisfied,

$$\lim_{x \to \infty} \frac{f'(x)(1 - F(x))}{f^2(x)} = -1, \tag{32}$$

where $f$ and $f'$ denote the probability density function of Erlang distribution and its derivative respectively. To this end, we have

$$f(x) = \frac{\lambda^B x^{B-1} e^{-\lambda x}}{(B - 1)!}, \quad f'(x) = \left( \frac{B - 1}{x} - \lambda \right) f(x).$$

After substitution in (32), the lemma follows. \hfill \blacksquare

Therefore, there exist constants $a_M > 0$ and $b_M$ so that $(X - b_M)/a_M$ converges in distribution to the standard Gumbel distribution, as $M \to \infty$. A possible choice for $a_M$ and $b_M$ is given
by (16). Specifically, \( b_M \) can be computed by solving the equation \( b_M = F^{-1}(1 - 1/M) \). Using the following representation of the Erlang distribution,

\[
F(x) = 1 - e^{-\lambda x} \left[ \frac{(\lambda x)^{B-1}}{(B-1)!} + \Theta(x^{B-2}) \right],
\]

it is obtained that,

\[
\frac{1}{M} = e^{-\lambda b_M} \left[ \frac{(\lambda b_M)^{B-1}}{(B-1)!} + \Theta(b_M^{B-2}) \right].
\]

By taking the logarithm of both sides of the equation, and using the relation \( \ln(r + t) = \ln r + \ln(1 + \frac{t}{r}) \), we have,

\[
\lambda b_M = \ln M + (B-1) \ln(\lambda b_M) - \ln(B-1)! + o(1),
\]

which yields the following expression for \( b_M \),

\[
b_M = \lambda^{-1}(\ln M + (B-1) \ln \ln M - \ln(B-1)!) .
\]

By substituting \( b_M \) in (16), it is then obtained that \( a_M = \lambda^{-1} \). Substituting \( a_M \) and \( b_M \) in (17), the following expression is obtained for \( \mathbb{E}[X] \),

\[
\mathbb{E}[X] = \frac{\ln W + (B-1) \ln \ln W + \Theta(B \ln B)}{-\ln p}, \tag{33}
\]

as \( W \to \infty \). Given the constant approximation error (31), it is obtained that \( \mathbb{E}[N_{FEC}] \in \Theta(\log W) \). Since \( B \) is constant, the number of per packet transmissions, i.e., \( \frac{1}{B} \mathbb{E}[N_{FEC}] \), also is in \( \Theta(\log W) \).

Using \( \mathbb{E}[N_{FEC}] \), we show that \( \mathbb{E}[Q_{FEC}] \in \Theta(W \log W) \). The proof is similar to the proof presented for \( \mathbb{E}[Q_{ARQ}] \) in the previous section, and hence only a sketch of the proof is presented in the following lemmas.

**Lemma 5.** \( \mathbb{E}[Q_{FEC}] \in O(W \log W) \).

**Proof:** In the worst-case, the coding block that requires the most number of packet transmissions is the block with the smallest sequence number in the window. For every packet transmission for this block, \( W - 1 \) other packets are transmitted for the proceeding blocks. Thus, the expected receiver’s buffer occupancy is upper bounded by \( W \log W \).

**Lemma 6.** \( \mathbb{E}[Q_{FEC}] \in \Omega(W \log W) \).
Proof: Assume that the coding block with the largest sequence number is at the end of the window. Thus, all coding blocks before it can be potentially delivered to the application if they are receiver at the receiver in-order. Consider the time it takes for the first $M/2$ coding blocks to be delivered to the application. In the best-case, these coding blocks arrive at the receiver in an increasing order. Thus, the best-case time to receive the first $M/2$ blocks is upper bounded by the time it takes to receive $W/2$ coded packets belonging to the first half of the window. In the previous section, we showed that this time is upper bounded by a constant. Thus, the expected receiver’s buffer occupancy is lower bounded by $W \log W$.

\[ \mathbb{E}[Q_{\text{FEC}}] \in \Theta(W \log W). \]

Proof: The theorem follows from Lemmas 5 and 6.

Scaling Regime II: Growing Coding Block Size

Recall that in this case, we have $B \in \Theta(W)$ so that $M$ is a constant. We show that, in this case, the expected number of transmissions per packet is constant. Consequently, we show that $\mathbb{E}[Q_{\text{FEC}}] \in \Theta(W)$. The proof relies on the Lebesgue’s Dominated Convergence Theorem stated below.

**Theorem 6** (Dominated Convergence Theorem). Suppose that $X_n \to X$ almost surely as $n \to \infty$. If there exists a random variable $Y$ having finite expectation and $|X_n| \leq Y$, for all $n$, then $\mathbb{E}[X_n] \to \mathbb{E}[X]$, as $n \to \infty$.

**Theorem 7.** $\mathbb{E}[N_{\text{FEC}}] \in \Theta(W)$.

Proof: Define $N$ as the number of transmissions per packet until the entire window is received at the receiver, that is, $N = \frac{1}{B}N_{\text{FEC}}$. Using the expansion introduced in (27), we are interested in characterizing $\mathbb{E}[N]$,

\[ \mathbb{E}[N] = \frac{1}{B} \mathbb{E}\left[ \max_{1 \leq i \leq M} N_i \right] = \mathbb{E}\left[ \max_{1 \leq i \leq M} \frac{N_i}{B} \right]. \] (34)

Now, we rewrite (34) as follows,

\[ \mathbb{E}[N_{\text{FEC}}] = \mathbb{E}\left[ \max_{1 \leq i \leq M} \frac{N_i}{B} \right] \\
= \mathbb{E}\left[ \max_{1 \leq i \leq M} \frac{N_{i1} + \cdots + N_{iB}}{B} \right]. \] (35)
TABLE I
ASYMPTOTIC PERFORMANCE OF SELECTIVE REPEAT PROTOCOLS.

| Protocol          | Throughput          | Reseq. Buffer | Reseq. Delay | Feedback Overhead |
|-------------------|---------------------|---------------|--------------|------------------|
| SR-ARQ            | $(1 - p)C$          | $\Theta(W \log W)$ | $\Theta(\log W)$ | $\Theta(\log W)$ |
| SR-FEC, $B = \Theta(1)$ | $(1 - p)C$          | $\Theta(W \log W)$ | $\Theta(\log W)$ | $\Theta(\log W)$ |
| SR-FEC, $B = \Theta(W)$ | $(1 - p)C$          | $\Theta(W)$    | $\Theta(1)$   | $\Theta(1)$      |

Note that the $N_i$’s are IID and hence, by applying the strong law of large numbers, $N_i/B \to 1/(1-p)$, as $B \to \infty$, almost surely. Thus, $N \to 1/(1-p)$ almost surely as $B \to \infty$. Define random variable $Y$ as $Y = \sum_{1 \leq i \leq M} N_i/B$. Notice that $N \leq Y$ for all values of $B$, and that $E[Y] = M/(1-p) < \infty$. Therefore, the Lebesgue’s Dominated Convergence Theorem can be applied, which yields,

$$\lim_{B \to \infty} E[N] = E\left[\max_{1 \leq i \leq M} \lim_{B \to \infty} \frac{N_{i1} + \cdots + N_{iB}}{B}\right],$$

$$= E\left[\max_{1 \leq i \leq M} \frac{1}{1 - p}\right] = \frac{1}{1 - p} = \Theta(1).$$

(36)

It then follows that $E[N_{\text{FEC}}] = B/(1 - p)$, which completes the proof.

**Theorem 8.** $E[Q_{\text{FEC}}] \in \Theta(W)$.

**Proof:** Notice that in this case, all coding blocks are decoded at the same time. Thus, the expected buffer occupancy is given by $W/(1 - p)$, which established the theorem.

**V. EXTENSIONS AND IMPLICATIONS**

**A. Lossy Feedback**

If an ACK packet is lost, the sender will time-out and retransmit a packet. As a result, the throughput achieved by the reliability protocols may suffer as some packets are unnecessarily transmitted due to lost ACKs. However, the receiver buffer requirement is not affected by lost ACKs as the packet corresponding to the lost ACK is buffered at the receiver like every other packet.

Assume that the ACK packets are lost with probability $p_a$. At the sender, a packet is considered lost if either the packet or its ACK is lost. With SR-ARQ, the sender retransmits the same packet after every time-out until it receives an ACK for that packet. Thus, the throughput achieved by
SR-ARQ is $\rho = (1 - p)(1 - p_a)C$. With SR-FEC, however, after every time-out a new coded packet is transmitted that could potentially be useful for decoding the corresponding block at the receiver. Only when the corresponding block is (or will be) full rank then time-outs are wasteful. Thus, SR-FEC potentially achieves higher throughput compared to SR-ARQ.

By using more sophisticated feedback mechanisms, however, one could improve the throughput of both protocols to $\rho = (1 - p)C$, without affecting the receiver buffer requirement. One such mechanism is described next. Our current protocols incur feedback overhead of $\Theta(\log W)$ per-packet (see subsection \ref{sec:feedback}). As long as the packet length grows faster than the per-packet feedback overhead, there will not be any loss in throughput asymptotically. To this end, we show that the receiver can send $(1 + \epsilon) \log W$ ACKs per each received packet, for any $\epsilon > 0$, and still maintain negligible feedback overhead if the packet size scales as $\omega(\log^2 W)$.

Suppose that the receiver sends $n$ ACKs back-to-back for every received packet. The achieved throughput is $\rho = (1 - p)(1 - p_a^n)C$. To achieve full throughput, the throughput loss $p_a^n C$ should converge to 0 as $W$ grows to infinity. That is, $W p_a^n \rightarrow 0$, as $W \rightarrow \infty$, or equivalently, $p_a^n = o\left(\frac{1}{W}\right)$, which yields,

$$n = \log_{p_a} o\left(\frac{1}{W}\right) = \Omega\left(\log_{p_a} \frac{1}{W^{1+\epsilon}}\right),$$

for any $\epsilon > 0$.

Specifically, transmitting $(1 + \epsilon) \log W$ redundant ACKs ensures asymptotically full throughput for $p_a < 2^{-1+\epsilon}$. For instance, for $p_a < 0.5$, which is the case of interest in this work (TCP does not work for loss probabilities higher than 10%), transmitting $\log W$ redundant ACKs suffices to achieve throughput $\rho = (1 - p)C$, as $W \rightarrow \infty$.

B. Dependent Coded Packets

In the preceding sections, we assumed that the sender keeps track of the coded packets transmitted to the receiver in order to avoid transmitting dependent coded packets. Alternatively, the sender might simply generate and transmit coded packets regardless of their dependency. This approach, while having the same buffer requirement, comes with a throughput penalty, as there is a non-zero probability that the received packets at the receiver are not independent. In this subsection, using results from the theory of random matrices \cite{16}, we characterize the throughput loss incurred due to dependent packets.
Let $N^*_i$ denote the number of transmitted coded packets until the receiver receives $B$ independent coded packets for coding block $i$. Let $N^*_{i_k}$ denote the number of transmissions until the receiver receives the $k$-th independent coded packet given that it has already received $k - 1$ independent coded packets for the coding block. We have,

$$N^*_i = N^*_{i_1} + N^*_{i_2} + \cdots + N^*_{i_B}. \quad (38)$$

There are $B$ packets in a coding block, and hence, there are $2^B$ subsets of packets that can be XORed to create coded packets. While waiting to receive the $k$-th independent coded packet, the receiver has already received $k - 1$ independent coded packets. Therefore, $2^{k-1}$ packets out of the $2^B$ possible coded packets will not be independent of the already received $k - 1$ packets. Let $\alpha_k$ denote the probability that a received packet is independent of the existing $k - 1$ packets. The following relation holds,

$$\alpha_k = \frac{2^B - 2^{k-1}}{2^B} = 1 - 2^{k-B} - 1. \quad (39)$$

A packet (dependent or independent) is received with probability $1 - p$ at the receiver. Therefore, $N^*_{i_k}$ is geometrically distributed with success probability $q_k = \alpha_k(1 - p)$. It then follows that,

$$\mathbb{E}[N^*_i] = \sum_{k=1}^{B} \frac{1}{q_k} = \frac{1}{1-p} \left( B + \sum_{k=1}^{B} \frac{1}{2^k-1} \right), \quad (40)$$

where the summation term quickly converges to a constant ($\approx 1.606695$) even for small values of $B$. This means that, on average, the receiver needs only two extra packets in order to be able to decode a block of packets. Specifically, let $\delta$ denote the number of extra packets (dependent or independent) that will be delivered to the receiver. Using our earlier argument, the probability that a newly arrived packet is independent of the previously received $k - 1$ independent packets is given by,

$$\beta_k = \frac{2^B - 2^{k-1}}{2^B+\delta} = 1 - 2^{k-B-\delta-1}. \quad (41)$$

Thus, the probability of receiving $B$ independent packets out of the $B + \delta$ received packets, denoted by $\pi(\delta)$, is given by,

$$\pi(\delta) = \prod_{k=1}^{B} \beta_k = \prod_{k=1}^{B} (1 - 2^{-k-\delta}), \quad (42)$$

where, we have $\pi(\delta) \geq 1 - \sum_{k=1}^{B} 2^{-k-\delta} \geq 1 - 2^{-\delta}$. By taking $\delta = \Theta(\log B)$, we have $\pi(\delta) \to 1$, as $B \to \infty$. Next, we compute the throughput loss because of dependent packets. Let $\Delta \rho$ denote
the throughput loss, we have:

\[ \Delta \rho = \frac{M \cdot \delta}{(1 - p) C} = \frac{R \cdot M \cdot \delta}{1 - p \cdot W}. \]  

(43)

Therefore, the throughput loss is given by,

\[ \Delta \rho = \begin{cases} 
\frac{R \cdot \log B}{1 - p \cdot W} = \Theta(1), & \text{if } B = \Theta(1), \\
\frac{R \cdot \log W}{1 - p \cdot W} = o(1), & \text{if } B = \Theta(W). 
\end{cases} \]  

(44)

C. Protocol Overhead

Our results for the re-sequencing buffer occupancy can be used as guidelines for dimensioning the sender and receiver buffer space. Specifically, for SR-ARQ and SR-FEC with \( B = \Theta(1) \), the proper amount of buffer space at the sender and receiver is \( \Theta(W \log W) \). On the other hand, when \( B = \Theta(W) \) under SR-FEC, the buffer space at the sender and receiver should be scaled as \( \Theta(W) \).

1) Data Channel Overhead: Both protocols require data packets in the sender and receiver buffers to have unique sequence numbers. Thus, they both incur the same overhead in the forward direction from the sender to the receiver. Moreover, the packet header overhead due to sequence numbers is given by \( \Theta(\log W) \).

2) Feedback Channel Overhead: Every packet has to be individually ACKed in SR-ARQ. Thus, each ACK packet carries the sequence number of the packet it is acknowledging. In SR-FEC, however, each ACK carries the sequence number of the corresponding coding block only. Thus, while both protocols send one ACK per every packet received at the receiver, the header overhead can be different for each protocol. Specifically, for SR-ARQ and SR-FEC with \( B = \Theta(1) \), the per ACK overhead is in \( \Theta(\log W) \). On the other hand, with \( B = \Theta(W) \), SR-FEC incurs only \( \Theta(1) \) header overhead per ACK.

D. Scaling Packet Length

To amortize the header overhead of \( \Theta(\log W) \), the packet length should also scale with the window size as \( \omega(\log W) \). This has implications for both the packet loss probability and throughput, as discussed below.
1) Effect on Packet Loss Probability: Let $L$ denote the packet length and $p_e$ denote the bit error rate (BER) at the physical layer. For the ease of exposition, we ignore the intermediate protocol layers and assume that application packets are passed directly to the physical layer for transmission. The packet loss probability $p$ is then given by,

$$p = 1 - (1 - p_e)^L = 1 - e^{L \ln(1 - p_e)} \approx 1 - e^{-Lp_e}. \quad (45)$$

Thus, as the packet length increases, the packet error probability approaches 1 exponentially fast. To compensate for the increased packet length in order to keep the packet error probability constant, one can decrease the coding rate at the physical layer by adding more error control data to each packet. The side effect of the decreased coding rate at the physical layer is a lower per-flow capacity at the application layer. Nevertheless, the asymptotic performance of SR-ARQ and SR-FEC in terms of buffer requirement remains unchanged.

2) Effect on Throughput: Consider a block coding scheme at the physical layer. Define the coding rate $r$ as $r = k/n$, where $n$ and $k$ ($n \geq k$) denote the length of coding blocks and information messages respectively. Since packets are directly passed to the physical layer, each message corresponds to a packet, i.e., $k = L$. For general classes of codes, it has been shown that the maximal coding rate achievable at block length $n$ with error probability $p$ is closely approximated by,

$$r = \frac{L}{n} \approx C - \sqrt{\frac{L}{n} Q^{-1}(p)}, \quad (46)$$

where $C$ is the capacity of the underlying channel, $V$ is a characteristic of the channel referred to as channel dispersion, and $Q$ is the complementary CDF of the standard normal distribution. Notice that, as $n \to \infty$, the maximal achievable rate approaches the channel capacity. Thus, a physical layer that is designed optimally, not only does not penalize throughput as the packet length grows but also exhibits even a better performance.

Let assume that our objective is to keep the loss probability constant at $p$ regardless of the packet length. Then, using the above approximation, it is obtained that $n = \Theta(L)$, which means that a constant throughput can be achieved for any packet length. Specifically, for any fixed packet length $L$, let assume that our objective is to achieve the throughput $\rho = (1 - p)C$. It is obtained that $n = V \cdot \left(\frac{Q^{-1}(p)}{pC}\right)^2$, indicating that a fixed packet loss probability at a constant throughput can be achieved by appropriately controlling the coding block length at the physical layer.
E. Implications for Protocol Design

The scaling results for various performance metrics are summarized in Table I. The conclusion is that SR-FEC with $B = \Theta(W)$ asymptotically outperforms SR-ARQ in terms of buffer requirement, delay and protocol overhead.

VI. RELATED WORK

The analysis of selective repeat protocols has received significant attention in the literature. Our primary goal in this paper is to characterize the asymptotic performance of selective repeat with ARQ and FEC, which has not been considered in the literature. A summary of some related works follows.

Selective Repeat ARQ: Some of the early work on the analysis of the selective repeat protocol with ARQ can be found in [10]–[13]. These classic works have been extended in several recent papers to consider more general systems (e.g., Markovian error models and packet arrival rates) [17]–[19], specifically in cellular networks in which selective repeat is typically implemented between base stations and user devices. All these works have considered exact or approximate analysis of selective repeat with ARQ.

Selective Repeat Hybrid ARQ: Hybrid ARQ schemes combine ARQ and FEC mechanisms similar to SR-FEC considered in this paper. The difference is that in hybrid ARQ, the coding rate is pre-determined based on the transmission environment. For instance, a block coding scheme such as Reed-Solomon codes is used to generate a fixed number of coded packets in order to cope with packet losses. In SR-FEC, however, coded packets are generated on-demand until the receiver successfully decodes the original packets (i.e., rateless coding). Several examples of analytical work on selective repeat with hybrid ARQ can be found in [20], [21] and references therein. Note that none of these works has considered the asymptotic behavior of hybrid ARQ schemes.

Packet Reordering Delay: The analysis of [22] considers reliable data transmission but assumes that variable network delay is the cause of packet re-ordering while the network is perfectly reliable in the sense that no packets are lost. Along the same line, in [23], it is assumed that the cause of variable delay is multi-path routing in the network. Their focus is on capturing the effect of packet reordering on receiver buffer, which is quite different from the lossy scenario considered in this paper.
Asymptotic Analysis: A large-deviation analysis of receiver buffer behavior under selective repeat was presented in [24]. Additionally, using standard results on coupon collector problem, the authors provide an asymptotic analysis of the receiver buffer behavior for ARQ mechanism, which is valid only when the loss rate is very high, as opposed to our results that characterize receiver buffer behavior under arbitrary loss rates for both ARQ and FEC mechanisms.

VII. Conclusion

In this paper, we studied reliable data transfer in high-speed networks, where a large number of flows can be accommodated at high transmission speeds. We focused on two sliding window mechanisms called SR-ARQ and SR-FEC based on ARQ and FEC, and characterized their performance in the asymptotic regime of large window sizes. Specifically, we showed that, while asymptotically achieving equal throughput, SR-FEC with a large coding block size achieves $\log W$ improvement in buffer requirement and delay compared to SR-ARQ. However, SR-FEC with a small coding block size is no advantageous compared to SR-ARQ. It would be interesting to study the performance of these protocols when the available buffer space is limited.

REFERENCES

[1] Y.-T. Li, D. Leith, and R. N. Shorten, “Experimental evaluation of TCP protocols for high-speed networks,” IEEE/ACM Trans. Netw., vol. 15, no. 5, Oct. 2007.
[2] G. Appenzeller, I. Keslassy, and N. McKeown, “Sizing router buffers,” in Proc. ACM SIGCOMM, Portland, USA, Aug. 2004.
[3] Y. Gu, D. Towsley, C. Hollot, and H. Zhang, “Congestion control for small buffer high bandwidth networks,” in Proc. IEEE Infocom, Anchorage, USA, May 2007.
[4] J. Kurose and K. Ross, Computer Networking: A Top-Down Approach. New Jersey, USA: Addision-Wesley, 2012.
[5] O. Tickoo et al., “LT-TCP: End-to-end framework to improve TCP performance over networks with lossy channels,” in Proc. IEEE IWQoS, Passau, Germany, Jun. 2005.
[6] J. K. Sundararajan et al., “Network coding meets TCP,” in Proc. IEEE Infocom, Rio de Janeiro, Brazil, Apr. 2009.
[7] D. MacKay, Information Theory, Inference, and Learning Algorithms. Cambridge, UK: Cambridge University Press, 2003.
[8] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, “ATP: A reliable transport protocol for ad-hoc networks,” in Proc. ACM Mobihoc, Annapolis, USA, Jun. 2003.
[9] E. Kohler, M. Handley, and S. Floyd, “Designing DCCP: Congestion control without reliability,” ACM SIGCOMM Comput. Communin. Rev., vol. 36, no. 4, pp. 27–38, Oct. 2006.
[10] D. Towsley and J. K. Wolf, “On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission schemes,” IEEE Trans. Commun., vol. 27, no. 4, pp. 693–702, Apr. 1979.
[11] A. Konheim, “A queueing analysis of two ARQ protocols,” *IEEE Trans. Commun.*, vol. 28, no. 7, 1980.

[12] M. E. Anagnostou and E. N. Protonotarios, “Performance analysis of the selective repeat ARQ protocol,” *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 127–135, Feb. 1986.

[13] Z. Rosberg and N. Shacham, “Resequencing delay and buffer occupancy under the selective-repeat ARQ,” *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 166–173, Jan. 1989.

[14] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics*. New Jersey, USA: Wiley, 1978.

[15] L. de Haan and A. Ferreira, *Extreme Value Theory: An Introduction*. New York, USA: Springer, 2006.

[16] C. Studholme and I. F. Blake, “Random matrices and codes for the erasure channel,” *Algorithmica*, vol. 56, no. 4, 2010.

[17] D.-L. Lu and J.-F. Chang, “Performance of ARQ protocols in nonindependent channel errors,” *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 721–730, May 1993.

[18] R. Fantacci, “Queueing analysis of the selective repeat automatic repeat request protocol wireless packet networks,” *IEEE Trans. Veh. Technol.*, vol. 45, no. 2, 1996.

[19] J. G. Kim and M. Krunz, “Delay analysis of selective repeat ARQ for a Markovian source over a wireless channel,” *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, 2000.

[20] S. Kallel, “Analysis of a type II hybrid ARQ scheme with code combining,” *IEEE Trans. Commun.*, vol. 38, no. 8, 1990.

[21] L. Badia, M. Levorato, and M. Zorzi, “Markov analysis of selective repeat type II hybrid ARQ using block codes,” *IEEE Trans. Commun.*, vol. 56, no. 9, 2008.

[22] R. Fantacci, “Analysis on packet resequencing for reliable network protocols,” *Performance Evaluation*, vol. 61, no. 4, 2005.

[23] K. Zheng, X. Jiao, M. Liu, and Z. Li, “An analysis of resequencing delay of reliable transmission protocols over multipath,” in *Proc. IEEE ICC*, Cape Town, South Africa, May 2010.

[24] K. D. Turck and S. Wittevrongel, “Receiver buffer behavior for the selective repeat protocol over a wireless channel: an exact and large-deviations analysis,” *Industrial and Management Optimization*, vol. 6, no. 3, 2010.