5775 Morehouse Drive
San Diego, CA 92121
(858) 587-1121

Why DF Raptor® is Better Than Reed-Solomon for Streaming Applications

Why Digital Fountain's DF Raptor™ Technology Is Better Than Reed-Solomon Erasure Codes For Streaming Applications

Digital Fountain's DF Raptor™ technology is the world's most powerful erasure correction code. The need for erasure correction is especially apparent in data packet networks where an entire packet of data can be considered lost ("erased") if it has been en route or if it has experienced so much data corruption that any error correction code that may be employed cannot fully correct the data. While bit-level error correction is fairly well-known among engineers, packet-level erasure correction may be relatively unfamiliar.

For a given application requiring bit-level error correction, many different codes and coding techniques are available, each with their own specific advantages and disadvantages. For applications requiring packet-level erasure correction and the recovery of lost packets, however, the available coding options may be less well understood. This paper highlights the advantages of Digital Fountain's patented DF Raptor FEC (forward error correction) technology when compared to application of the well-known Reed-Solomon code as a packet-level erasure correction code.

In streaming applications, the effects of even small levels of packet loss can be readily apparent to the end user. In a head-to-head comparison of DF Raptor and Reed-Solomon erasure codes as used to protect streaming applications, the following conclusions can be drawn:

- A DF Raptor code provides exceptional flexibility, while Reed-Solomon erasure codes are subject to constraints that limit their utility and diminish their relative performance

- A DF Raptor code protects against packet loss with greater efficiency (more protection with less overhead) than do comparable Reed-Solomon erasure codes

- A DF Raptor code requires significantly less processing power than Reed-Solomon erasure codes for encoding or decoding, and the required processing power for DF Raptor FEC increases linearly with the level of provided protection, not quadratically as it does for Reed-Solomon erasure codes

- A DF Raptor code allows a given application to be optimally addressed in terms of the degree of packet loss protection, bandwidth expansion, and processing demands, while Reed-Solomon erasure codes, because of their relative inflexibility, incur a performance cost in one dimension when optimized in another

Packet-Level FEC for Streaming Applications

In streaming applications, a continuous flow of data packets is generated by the transmitter for delivery to one or more receivers. Both server-client and peer-to-peer network architectures can employ streaming, and streaming applications can be broadly defined to encompass text streaming and telemetry as well as video and audio. The term "streaming," however, especially relates to isochronous processes such as video and audio where the data needs to be processed and output at the receiver at the same rate of flow as generated by the source.

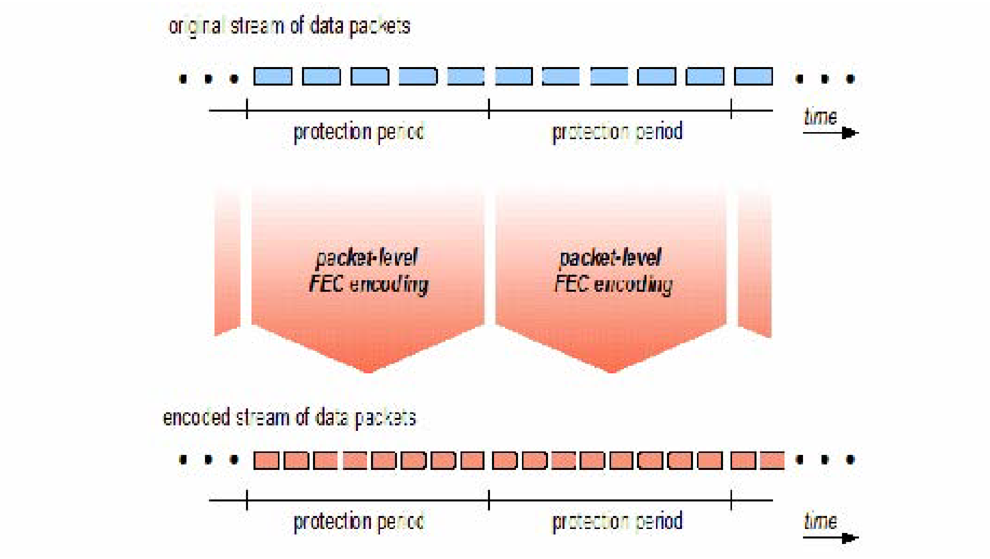
Services such as video-on-demand, 2-way video conferencing, IPTV, IP radio, in-home media distribution, and mobile TV all utilize streaming as a way to deliver content. Streaming applications thus embody a wide range of data rates, expected quality levels, transmission media, networking techniques, and receiver types. For example, a mobile streaming application may be a low-data-rate, moderate-quality, narrowband wireless broadcast to handheld devices with modest processing power, while an IPTV streaming application may be a high-data-rate, high-quality, broadband unicast transmission over xDSL or fiber-to-the-home to a consumer-grade set-top box.

If any packets of a stream are lost or irreparably corrupted during transmission, however, then information will be missing at the receiver. The video compression techniques (MPEG-2, MPEG-4/H.264, or WMV9/VC-1) used for streaming video are particularly sensitive to the loss of any packets in transmission. If not recovered, missing packets – or even a single missing packet -- of a video stream can result in garbled audio as well as frozen frames, frame skips, macroblock errors, or other visible or audible distortions and artifacts.

Missing packets can be re-transmitted, but any such retransmission of the data will be perceived as an interruption of the data flow unless the data has been adequately buffered at the receiver. Even with buffering, high instantaneous levels of packet loss may cause the buffer to be emptied while waiting for successful re-transmissions, thus producing a perceptible interruption or glitch in the playback of the stream. In addition, if a streaming server is used to send one or many different streams to many receivers, the retransmission of lost packets for each individual receiver quickly becomes impractical.

The use of packet-level FEC technology provides a solution. By employing an FEC erasure code to protect the data stream, lost packets can be independently recovered at each receiver without requiring any retransmission of data. As shown in Figure 1, erasure correction coding for streaming applications can be applied to consecutive equal-size blocks of the original data stream. Each block – constituting a protection period in time – is encoded and protected separately and then transmitted as part of a new encoded stream. In this way, the original stream is replaced by an encoded stream that is protected against packet loss.

Figure 1 Packet-level FEC encoding of a data stream


original stream of data packets
protection period
protection period
time
packet-level FEC encoding
packet-level FEC encoding
encoded stream of data packets
protection period
protection period
time

FEC coding deliberately introduces some redundancy, creating more packets than were originally present during the protection period in order to compensate for potential packet loss. The number of additional packets – the protection amount -- produced by the encoding process depends on what maximum amount of packet loss is to be protected and on the specific erasure code. This overhead directly translates into bandwidth expansion -- to maintain the timing of the original stream, the encoded stream of data packets must be transmitted over the same duration as the protection period, requiring a faster data rate than the original stream in order to accommodate the additional packets.

Comparing different FEC erasure codes in a streaming application thus involves consideration of how well each FEC code protects against packet loss for a given amount of bandwidth expansion, subject to the available processing power at the sender and receiver. As will be shown here, Digital Fountain’s DF Raptor code can maximize packet loss resiliency while minimizing bandwidth expansion and processing requirements, while, by contrast, Reed-Solomon erasure codes are unable to optimize all three aspects at once.

Erasure Codes

In block FEC erasure codes, a source block of k source symbols is encoded as$n > k$ output symbols that are transmitted to a receiver so that the original k source symbols can be recovered even if some of the output symbols are not received. This recovery and protection against symbol loss requires no additional data from the transmitter aside from a small amount of header information to allow the output symbols in each encoded packet to be identified by the receiver. The code rate is defined as k/n , inversely reflecting the degree of redundancy introduced by the additional symbols transmitted to the receiver. The original source symbols can then be recovered solely on the basis of those output symbols that are correctly received by the receiver. The process of erasure correction is distinct and different from error correction – in error correction, all the output symbols are received but some might be corrupted, requiring the error-correction process to identify the corrupted symbols and then recover and correct the original k source symbols; in erasure correction, by contrast, all the received output symbols are correct but some might be missing, requiring the erasure correction process to recover the missing original k source symbols from the received output symbols.

Reed-Solomon Erasure Codes

Reed-Solomon codes are familiar to engineers from their use as error correction codes in such technologies as Compact Discs (CDs), Digital Video Discs (DVDs), xDSL, digital cellular telephony, and satellite communications. In the case of a CD or DVD player, for example, the "transmitter" is the data as recorded as pits on the disc, and the "receiver" is the sensor and its associated processing used to read the data reflected or dispersed off the disc by the player's laser. In such applications, the complexity of the Reed-Solomon error correction algorithm typically requires that the decoder be implemented in hardware because of the relatively demanding processing requirements.

Reed-Solomon codes, when used as error correction codes, are well-known to be capable of correcting any combination of $\lfloor (n-k)/2 \rfloor$ or fewer errors (where the symbol "$\lfloor . \rfloor$", when operating on any variable x , denotes the largest integer not to exceed x). By contrast, Reed-Solomon codes, when used as erasure codes, are capable of correcting $(n - \mathrm{k})$ erasures from any successfully received set of
k symbols.

Reed-Solomon codes are powerful linear block error correction codes, but inefficiencies and limitations can be evident in their use as packet-level erasure codes. The fundamental definition of a Reed-Solomon code imposes a practical limitation: the Reed-Solomon algorithm requires operations over an extended Galois field, but to be tractable the field element size is typically limited to 8 bits (one byte), especially if the implementation is to be software-based or if processing power is limited. As a consequence, with single-byte field elements, the values of k and n for Reed-Solomon codes are constrained such that $0 < k < n \leq 255$.

The values of k and n may be further constrained by processing limitations: for Reed-Solomon codes, the number of operations required for either encoding or decoding grows quadratically with the source block size k and is related to the value of $n-k$. As a consequence, the protection of large amounts of data (large source blocks) against high levels of packet loss (large values of $n-k$) can be limited by the available processing power.

The way in which packets are constructed into a source block also directly affects the performance of a Reed-Solomon erasure code in protecting against packet loss. If fixed-length packets of size P bytes are to be protected, then a single Reed-Solomon code with a symbol size of P bytes can be used for a maximum amount of data equal to$k \times P$ bytes. To protect more than this amount, the data must be segmented into multiple blocks, each block protected by a different Reed-Solomon code, and the encoded blocks interleaved.

Similarly, if variable-length packets are to be protected, then the source block must be constructed so that a symbol size of S single-byte field elements allows a single Reed-Solomon code to be applied to the same set of packets. The maximum amount of data that can be protected by a single Reed-Solomon code is then $k \times S$, and, as with fixed-length packets, if more than this amount of data must be protected, then the data must be segmented in multiple blocks, each block protected by a different Reed-Solomon code, and the encoded blocks interleaved.

When more than one Reed-Solomon code is used and interleaved – either because a large amount of data must be protected or because variable-length packets require segmentation of the data into multiple blocks – then performance can deteriorate because of the randomly distributed nature of packet loss. While a single large block of encoded data will experience actual packet losses close to the average packet loss rate, multiple smaller blocks of encoded data are each more likely to experience actual packet loss levels that are either higher or lower than the average loss rate. Because the data has been segmented into multiple blocks and each block independently protected by a Reed-Solomon code, more data must be transmitted using interleaved short blocks in order to provide the same level of protection as would be needed if a single Reed-Solomon code could be applied over all the input data as a single large block, where this additional data represents interleaving overhead. Interleaving overhead is a key reason why Reed-Solomon erasure codes reveal inferior performance in many practical applications – a DF Raptor code, by contrast, does not require any such segmentation into smaller blocks and thus does not incur any interleaving overhead.

The choice of symbol size and how the source block is constructed can also affect Reed-Solomon erasure code performance. The larger the symbol size is, then the more data that can be protected as a block and the easier it may be to avoid interleaving Reed-Solomon codes. In the case of variable-length packets, however, a large symbol size generally entails the use of more padding bits than with a small symbol size, implying greater inefficiency due to the processing that must be expended to encode and decode the padded bits and the effective overhead associated with transmitting redundant packets that are to some degree serving to correct the padded bits.

In general, a number of trade-offs must be considered in order to use Reed-Solomon codes as erasure codes. A particular Reed-Solomon code, designated R-S(n,k,S), is specified by the value of n , the number of output symbols; the value of k , the number of source symbols; and the value of S , the symbol size in bytes. For a given application, the values of n , k and S , which generally determine the protection period and protection amount, must be chosen after taking into account:

- The desired objective to correct up to $n-k$ erasures of symbols of size S so that the maximum symbol loss that the transmission of n symbols (codewords) can tolerate is $(n-k)/n$
- The tolerable amount of bandwidth expansion n/k
- The processing burden associated with high values of k and $n-k$
- The need to adjust k and the symbol size S so that efficient construction of the source block involves minimal amounts of padding

Assessing these trade-offs for Reed-Solomon erasure codes is complicated by the practical constraint that $k < n \leq 255$ -- the flexibility is limited. Moreover, in streaming applications without any feedback from the receiver to the transmitter, these trade-offs must be assessed in advance and must thus capture the expected worst-case channel conditions. As it turns out, the overall processing requirements associated with Reed-Solomon codes and the constraints imposed on the values of n and k lead to clear disadvantages in the use of Reed-Solomon erasure codes when compared to a Raptor code.

DF Raptor Erasure Coding Technology

Digital Fountain's DF Raptor technology has been designed and optimized as an erasure correction code. A DF Raptor code with k source symbols that produces n output symbols using symbols of size S bytes is designated Raptor($n,,kS$). A DF Raptor code is a fountain code, capable of producing an unlimited sequence of encoded symbols (i.e., $n \rightarrow \infty$) from a block of k fixed-length source symbols.

Such codes are “rateless,” allowing the actual number of encoded symbols and thus the code rate to be determined as needed to combat the current level of network packet loss.

The source symbols used for a DF Raptor code can be complete data packets or some fraction of a packet. In general, a DF Raptor code can be constructed with any combination of the number of source symbols k and symbol size S such that $k \times S$ equals the total amount of data to be protected as a block, subject only to the constraint that$S$ must be less than or equal to the maximum packet size so that the loss of a packet always affects a whole number of symbols. If the original packets are of variable length, the source block can still be easily constructed, but it may be necessary to pad the last source symbol from each packet with zeroes. Typically, one chooses the source symbol to be small enough so that the inefficiencies associated with padding are minimized. The number of operations required for either encoding or decoding a DF Raptor code grows linearly with the size of the source block (i.e., processing complexity grows with the data), allowing the choice of code parameters (number of output symbols n , number of source symbols k , and symbol size S) to be relatively unconstrained by processing requirements.

From a source block of k source symbols, a DF Raptor code, as a fountain code, can produce any number of output symbols. Each output symbol is determined by the bit-wise XOR of a number of source symbols according to the DF Raptor algorithm, and each output symbol is thus the same size as a source symbol. The generated output symbols are then packetized as necessary for transmission over the network. At the receiver, a DF Raptor decoder is able to recover the source block from any set of k' received output symbols, where k' is slightly greater than k , the number of source symbols in the source block.

Note that the specific values of “ n ”, “ k ”, and “ S ” used for a Reed-Solomon and a DF Raptor code may be different for a given application. With a specific Reed-Solomon erasure code R-S(n,k,S), as many as$n-k$ erasures can be corrected, and the maximum loss that the transmission of n codewords can tolerate is $(n-k)/n$. With a DF Raptor erasure code, on the other hand, $k' > k$ received symbols must be used to recover the k source symbols, implying that the maximum loss that the transmission of
n output symbols can tolerate is $(n-k')/n$. Let $k' = (1+\varepsilon)k$, where $\varepsilon > 0$ represents the reception overhead associated with a DF Raptor code. A DF Raptor code has the property that a specific probability of decoding success is determined by the absolute number of additional symbols $(k'-k)$ that are received, so the reception overhead $\varepsilon = (k'-k)/k$ depends on the value of k and the desired probability that the source block can be fully recovered from the received output symbols. For Digital Fountain’s DF Raptor code, the reception overhead ε is typically less than 1%.

Unlike Reed-Solomon codes, however, a DF Raptor code does not have any limitation to the amount of data that can be protected: the number of source symbols k for a DF Raptor code can be as large as desired, and the symbol size S is not subject to any constraints other than that it be less than or equal to the packet size. As a result, the value of S can be much smaller for a DF Raptor code than is possible for a Reed-Solomon erasure code. There is also little penalty in the required processing power --a DF Raptor code's processing requirements are significantly less than that of a Reed-Solomon erasure code and grow only linearly with the source block size, while a Reed-Solomon erasure code's processing requirements grow quadratically with source block size.

In addition, a DF Raptor code, because it is a fountain code, allows the possibility of system designs that, in effect, adaptively adjust the code rate to protect against the actual amount of experienced packet loss. For example, the output symbols of a DF Raptor code for a source block of size k can be continually generated and transmitted until enough have been received to fully recover the original source block, at which time the receiver can transmit an acknowledgement to the transmitter requesting that transmission cease or that the next source block be transmitted. While a Reed-Solomon erasure code must generally be explicitly designed with a specific fixed code ratek/n for anticipated worst-case channel conditions, a DF Raptor code can be used either with a fixed code rate or by dynamically adapting to channel conditions by varying n and thus k/n . With this second approach, the bandwidth expansion associated with a DF Raptor code can be held to a minimum while still providing complete protection against packet loss.

Performance

Reed-Solomon and DF Raptor erasure codes exhibit markedly different performance. The erasure correction performance of Reed-Solomon erasure codes in streaming applications is directly affected by the inefficiencies associated with the limited number of symbols that can be used by the Reed-Solomon algorithm, while Reed-Solomon erasure codes can require an order of magnitude greater processing power for encoding and decoding than a DF Raptor code.

In streaming applications, the length of the protection period reflects a trade-off between robustness and latency. In general, the longer the protection period, the more likely that the actual number of lost packets over that period will approach the average packet loss rate so that a fixed code rate erasure correction code can be sure to provide adequate protection. At the same time, decoding incurs latency equal to the length of the protection period plus the processing time required for decoding – the received data must be buffered until all lost packets have been recovered through decoding if interruptions and glitches in the output stream are to be avoided.

Limitations to the Protection Period with Reed-Solomon Erasure Coding

Different streaming applications will have different latency requirements and packet loss protection needs. With a DF Raptor code, one can arbitrarily choose any protection period and latency without constraint. With Reed-Solomon codes, however, the protection period that can be supported by a single Reed-Solomon code is limited. Otherwise, a protection period of data must be segmented into multiple blocks that are protected individually, and the resulting encoded blocks are then interleaved, at the cost of additional processing as well as reduced packet loss protection.

In order to use a single Reed-Solomon code to protect a streaming application, the protection period is limited by the product of the symbol size$S$ and the number of source symbols k . With fixed-length packets, the packet length (or smaller) can be chosen as the symbol size. With variable-length packets, the distribution of packet sizes determines how the packets can be arranged and what number of bytes can be grouped as a symbol with a minimal amount of padding. The choice of symbol size is constrained, however, as it also determines the maximum size block of data and thus the maximum protection period that can be protected by a single Reed-Solomon code.

For example, assume that a Reed-Solomon source block is to be constructed with variable-length packets and that, due to padding considerations, a symbol size of 160 bytes is appropriate to minimize padding overhead. If a rate 3/4 erasure code R-S(255,192,160) is used, then the total amount of data that can be protected by a single code is 192 x 160 (30,720) bytes. The maximum protection period then depends on the specific media streaming rate: in this example, a media streaming rate of 5 Mbps would only allow a protection period as long as 49 milliseconds with a single Reed-Solomon code. If, however, 1024-byte symbols is used, then the total amount of data that can be protected by a single R-S(255,192,1024) code is 1024 x 192 (196,608) bytes and, for a media rate of 5 Mbps, a protection period as long as 315 milliseconds is then possible using a single code. With the larger symbol size, however, encoding the variable-length packet stream may incur excessive padding overhead.

If longer protection periods are desired, Reed-Solomon erasure codes can support them by segmenting the data in each protection period into multiple blocks, protecting each block with a different Reed-Solomon code, and interleaving. Additional complexity is associated with such interleaved Reed-Solomon erasure coding, and this approach also increases the likelihood that the percentage of packet loss actually experienced by one or more of the smaller blocks will exceed the level of loss experienced over the full protection period. The resulting degradation to performance can be significant.

The Effect of Interleaving Reed-Solomon Codes on Streaming Performance

The impact of interleaving on Reed-Solomon code performance can be seen by comparing the protection provided by interleaved Reed-Solomon erasure codes to that provided by a DF Raptor code. In streaming applications, packet loss, unless corrected by FEC, can result in glitches or other artifacts in the delivered audio or video. Even with FEC, the random nature of packet loss means that there is always a chance that the actual number of received packets is such that the employed erasure code will not have enough data to recover the original packets – there is always a probability of decoding error. Analysis or simulation can reveal the mean time between artifacts in the presence of any given distribution of packet loss, where the variability of the packet loss errors results in a shorter and thus worse mean time between artifacts for interleaved Reed-Solomon erasure codes than for a DF Raptor code.

Figure 2 shows a specific example to highlight the relative mean time between artifacts as a function of randomly distributed packet loss for several Reed-Solomon and DF Raptor codes. In Figure 2, a media rate of 5 Mbps as might be appropriate for a video stream delivered via IPTV is protected using a 500millisecond protection period to minimize latency but ensure good picture quality. In this example, the data packets of the original stream are assumed to be fixed-length, allowing 1024 bytes to be used as the symbol size without padding overhead penalty. The protection amount – the additional overhead in the form of extra packets and bandwidth – has been held constant for the codes at a low value of approximately 4%.

The Reed-Solomon code parameters n and k have been chosen so that the data associated with a protection period of 500 ms (312,500 bytes) can be easily segmented and interleaved. Four different Reed-Solomon codes are considered: R-S(169,163,1024), R-S(113,109,1024), R-S(57,55,1024), and R-S(32,30,1024). With k equal to 163, up to 166,912 bytes can then be protected by a single Reed-Solomon code. To handle the full 500-millisecond protection period, two such blocks must be used, each protected by a different Reed-Solomon code and interleaved. With the smaller values of k employed by the other Reed-Solomon erasure codes, more than two blocks are required, resulting in further degradations to performance.

For a DF Raptor code, there are no constraints on the source block size, and a single DF Raptor code can always be used to protect as much data as necessary without requiring any similar segmentation and interleaving. A symbol size S of 256 bytes has been chosen here to minimize the DF Raptor reception overhead, and the parameters n and k have been selected as 1354 and 1306, respectively, so that all the data within the protection period is covered and the protection amount is the same as for the four Reed-Solomon erasure codes.

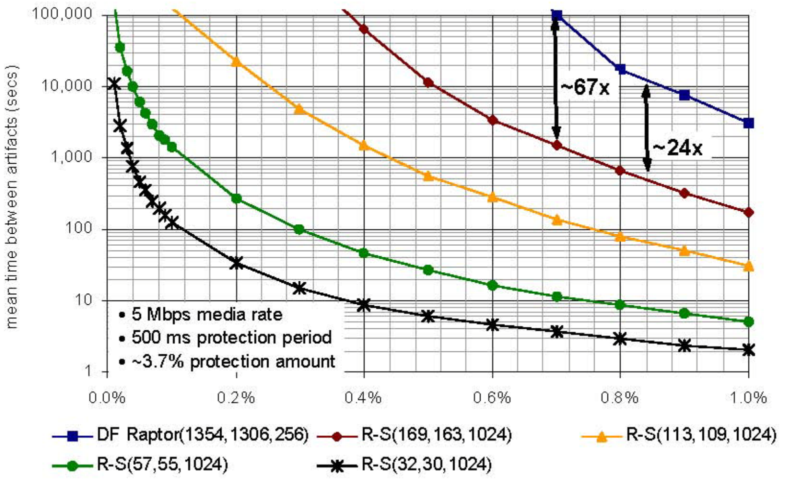


Figure 2 A DF Raptor code can provide a mean time between artifacts at least an order of magnitude greater than that provided by Reed-Solomon erasure codes

As can be seen from Figure 2, the DF Raptor code exhibits superior performance compared to the Reed-Solomon erasure codes – for a given rate of packet loss, the mean time between artifacts for the DF Raptor code is significantly greater than that of the Reed-Solomon codes. For example, the mean time between artifacts for the DF Raptor code is from ~24 to 67 times greater than the most powerful Reed-Solomon code shown here, the R-S(169,163,1024) code. The other Reed-Solomon codes have less erasure-correcting power and each show progressively worse performance than the DF Raptor code, primarily as a consequence of the additional blocks that must be interleaved to protect the 500 ms protection period.

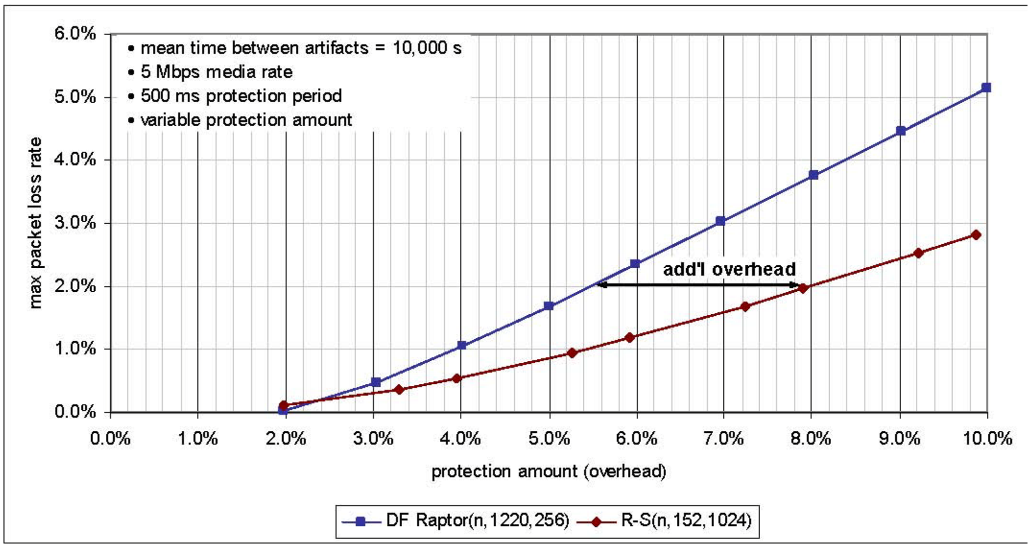
The difference in performance between the best Reed-Solomon erasure code shown here and the DF Raptor code is considerable and would be clearly evident to end users. For example, at a 0.8% packet loss rate, a video stream protected by the R-S(169,163,1024) code will exhibit a glitch or artifact on average every ~665 seconds (approximately every 11 minutes), while a stream protected by the DF Raptor code under the same conditions will show an artifact on average every ~17,230 seconds (approximately every 4 hours and 47 minutes).

Interleaving Overhead

Figure 3 provides a more general look at the phenomenon depicted in Figure 2. As in the case of Figure 2, Figure 3 considers a video stream with a media rate of 5 Mbps protected using a 500millisecond protection period. As before, interleaved Reed-Solomon codes are necessary to protect the data over this protection period. Here, however, with the symbol size and number of source symbols *k* held constant, different values of the number of output symbols *n* are used for the DF Raptor and ReedSolomon codes to vary the protection amount. Figure 3 then shows the maximum randomly distributed packet loss rate that each code can tolerate with a mean time between artifacts of at least 10,000 seconds (approximately every 2 hours and 46 minutes), where this level of performance is considered by some to be a minimum requirement for commercial streaming video services.

Figure 3 A DF Raptor code can tolerate more loss than a Reed-Solomon erasure code with the same overhead and mean time between artifacts

Figure 3 reveals that the maximum packet loss rate associated with a fixed mean time between artifacts is consistently higher for the DF Raptor code than for the Reed-Solomon code. In other words, the DF Raptor code provides greater protection against packet loss than the Reed-Solomon erasure code with the same amount of overhead and with the same mean time between artifacts.



The results of Figure 3 can also be interpreted by looking at what additional protection amount is needed for the Reed-Solomon erasure code to provide the same level of protection as the DF Raptor code. For example, in order to provide the same protection as the DF Raptor code against as much as 2% packet loss with a mean time between artifacts of 10,000 seconds, the Reed-Solomon code requires ~44% more additional overhead. As the required maximum level of protection increases, the additional overhead needed for the Reed-Solomon code to match a DF Raptor code's performance increases as well. This difference in overheads required for the same performance reflects the interleaving overhead associated with the Reed-Solomon erasure code as well as the ability of a DF Raptor code to use any combination of symbol and source block sizes.

As shown in Figures 2 and 3, the performance of Reed-Solomon erasure codes is worse than that of a DF Raptor code as a result of the block segmentation and code interleaving necessary to encode the entire desired protection period. Because several interleaved Reed-Solomon codes applied to different blocks are needed to protect a single protection period, each output block is subject to random variation in packet loss over its shorter duration. In these practical examples, the constraints imposed by Reed-Solomon erasure codes are strikingly visible to the end user as artifacts and corruptions to the delivered content that occur more frequently or at lower levels of packet loss than they would if a DF Raptor code were employed.

Theoretical Processing Requirements

Erasure codes can typically be implemented as software/firmware operating on general-purpose processors without the need for dedicated hardware. Performance depends on many variables, however, especially processor and memory access speeds. The processing demands associated with a particular code determine the maximum level of packet loss protection that can be provided and the maximum speeds at which data can be encoded and decoded, where these speeds are especially important for streaming applications in which decoding and, for 2-way interactive applications, encoding times add to latency. Ultimately, higher processing requirements for a particular FEC technology translate into higher costs.

While encoding and decoding of a full Reed-Solomon erasure codeword is quadratic with respect to the block size, encoding and decoding of a DF Raptor code is linear. To assess the relative processing requirements of Reed-Solomon and DF Raptor erasure codes, consider the theoretical number of XOR operations required for encoding and decoding. The workload per output symbol provides a convenient metric: in the case of encoding, the workload is the total number of XOR operations divided by n , the number of output encoded symbols; in the case of decoding, the workload is the total number of XOR operations divided by k , the number of output recovered source symbols.

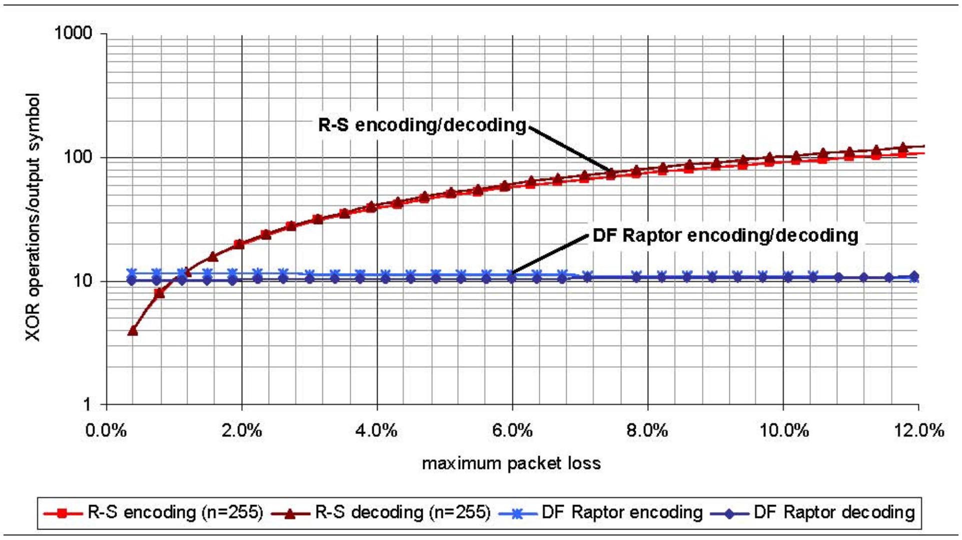
For Reed-Solomon erasure codes, assuming 8-bit field elements, the encoding workload can be approximated as $(k/n)\times (n-k)\times 4$ XORs per output encoded symbol, while the decoding workload is approximately $(n-k)\times 4$ XORs per output recovered source symbol.

For a DF Raptor code, a pessimistic approximation to the encoding workload of a Digital Fountain DF Raptor code is$4.5+ 7.5\times (k/n)$ XORs per output encoded symbol, while the decoding workload is approximately $10+ 4.5\times \min(n-k,k)/k$ XORs per output recovered source symbol (where the symbol “ $\min(x, y)$” denotes the minimum of the two variables x and y).

Figure 4 compares theoretical Reed-Solomon and DF Raptor workloads, neglecting any processing aspects that might be associated with Reed-Solomon interleaving. Encoding and decoding workloads are shown as a function of the maximum packet loss rate associated with different code rates: low maximum packet loss rates correspond to high code rates, and high maximum packet loss rates correspond to low code rates. For the Reed-Solomon erasure code, the value of n has been set to 255, and the value of k varied to determine the maximum packet loss L according to $L = (255- k)/255$. For the DF Raptor code, $k' > k$ output symbols must be received in order to recover lost packets, where the probability of successful decoding increases as k' increases. In particular, $k' - k =$ 12 additional symbols correspond to a success probability of 99.9%. Here, using the same value of k as for the Reed-Solomon code, the DF Raptor value of n has then been adjusted as needed for the given level of maximum packet loss L according to $k' = k + 12$ and $n = \lfloor k'/(1- L) \rfloor$.

Figure 4 DF Raptor encoding/decoding workloads are constant as packet loss protection increases; Reed-Solomon erasure code workloads increase rapidly with loss protection

As shown in Figure 4, the number of XORs per output symbol is nearly constant for encoding and decoding of a DF Raptor code, while the corresponding workload for Reed-Solomon erasure codes increases as the maximum allowed packet loss increases and the code rate decreases. At the very lowest levels of maximum packet loss corresponding to Reed-Solomon codes where k is between 252 and 254, the workloads for the Reed-Solomon and DF Raptor erasure codes are comparable. As soon as the maximum packet loss supported by the codes exceeds 2% and for all Reed-Solomon codes with $k < 250$, however, then the workload associated with Reed-Solomon erasure codes increases rapidly: from ~2x that of a DF Raptor code for a maximum packet loss of 2% to greater than 10x a DF Raptor code for maximum packet losses greater than 10%.



In many practical applications, the code parameters defining an erasure code must be chosen in advance in order to anticipate a worst-case level of loss. As a result, the maximum packet loss for which the code is designed can be expected to be 2% or greater, and the processing requirements of Reed-Solomon erasure codes can be expected to be higher than those of a DF Raptor code.

Trade-Offs in Packet Loss Protection and Processing Requirements

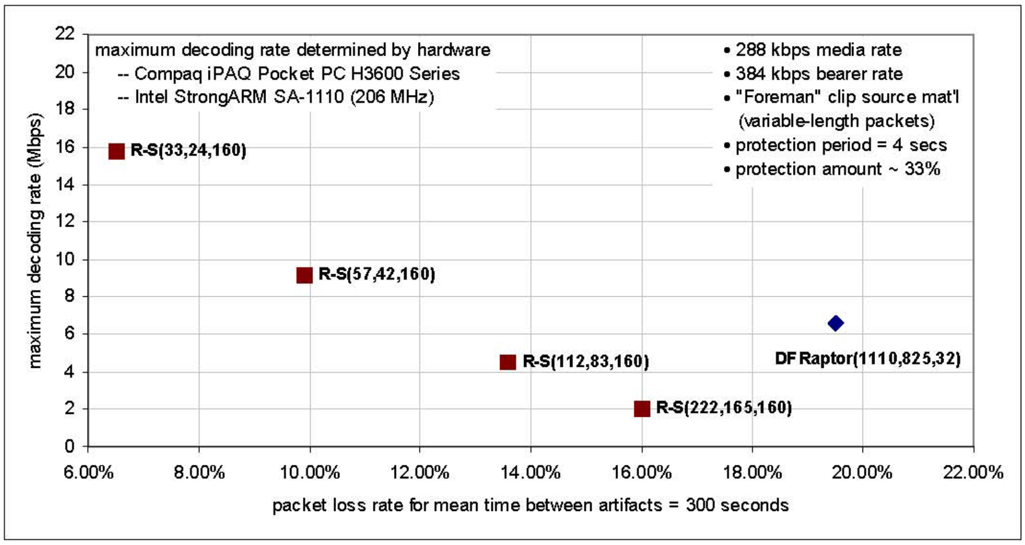
The performance advantage of a DF Raptor code over Reed-Solomon erasure codes becomes most evident when processing requirements are considered in the context of an actual implementation. Consider a digital cellular video-on-demand service where relatively lightweight clients must decode a low-rate 288 kbps video stream that has been protected by a rate 3/4 erasure code using a protection period of 4 seconds. Here, a Compaq iPAQ Pocket PC H3600 Series handheld device has been assumed to be representative, where this device employs an Intel StrongARM SA-1110 32-bit RISC microprocessor, operating at 206 MHz and capable of 235 Dhrystone 2.1 MIPS at that speed.

In order to assess processing requirements, the maximum decoding speed that can be supported by the StrongARM microprocessor when operating at 100% utilization has been evaluated. The processing power of a device used for streaming applications is not, of course, dedicated to FEC decoding – audio/video decoding and all the other applications hosted by the device must be able to run concurrently. The CPU cycles available for FEC decoding should thus be a small fraction of a device’s total processing capacity, but testing the maximum media rate that can be decoded at 100% utilization provides a uniform basis to compare processing requirements among different erasure codes.

Figure 5 compares the performance of several rate 3/4 erasure codes in such a scenario, using the Foreman clip that is often employed as a reference sample in MPEG-2 studies as the streaming content. For the Reed-Solomon codes, this variable-length packet stream was protected by segmenting it into 160-byte symbols. For the DF Raptor code, a smaller symbol size and larger source block size k than used with the Reed-Solomon codes are able to protect all the data over the protection period.

In a wireless environment, severe packet loss conditions are likely, and the mean time between artifacts represents a threshold of user acceptance as opposed to a desired level of delivered quality. On the x-axis of Figure 5, the maximum packet loss that can be tolerated before the mean time between artifacts equals 300 seconds is shown; on the y-axis, the maximum media rate that could be decoded by the iPAQ/StrongARM device operating at 100% utilization is shown. As can be seen, the DF Raptor code is closest to the upper right corner of the graph, indicating that it can tolerate the highest amount of packet loss before exhibiting artifacts and has the greatest amount of processing headroom.

Figure 5 A DF



s
a superior trade-off between packet loss protection and processing requirements than comparable Reed-Solomon erasure codes

In Figure 5, the trade-offs associated with Reed-Solomon erasure codes are clearly evident. The smallest Reed-Solomon code tested, R-S(33,24,160), requires the least processing power and thus allows the highest maximum decoding rate, but it is at the same time more susceptible to low packet loss rates than the other codes because of the low erasure-correcting power of its code parameters and the effects of interleaving overhead. As the Reed-Solomon codes become larger (with higher values for n and k), the required processing power increases as the ability to tolerate packet loss increases. The two larger and more powerful Reed-Solomon erasure codes require more processing power than the DF Raptor code, yet neither they nor the other Reed-Solomon codes are able to protect against as much packet loss as the DF Raptor code. A fundamental trade-off involved with Reed-Solomon erasure codes is evident here: the larger and more powerful Reed-Solomon erasure codes are better able to provide packet loss protection than smaller codes, but they require more processing power and may not be an economical choice given the target platform.

Optimizing In Different Dimensions

The ability to optimally select a code for a given application is more limited for Reed-Solomon erasure codes than for a DF Raptor code. For Reed-Solomon erasure codes, the constraint on the number of output symbols and the quadratic growth of processing complexity as the source block size increases lead to a complicated set of trade-offs:

- A long protection period increases the packet loss protection that can be provided by an erasure code in streaming applications, but the protection period that can be covered by a single Reed-Solomon erasure code is limited
- Long protection periods may require either or both a large symbol size and a large source block size in order to be covered by a single Reed-Solomon erasure code, but large symbol sizes introduce greater padding inefficiencies and large source block sizes imply high complexity
- If the protection period cannot be fully covered by a single Reed-Solomon code, interleaving can be used, but interleaving reduces the overall robustness to packet loss
- If interleaving must be used, then additional overhead and bandwidth expansion are required in order to achieve the same protection against packet loss as a comparable DF Raptor code, where additional overhead also implies some additional processing demand

For a DF Raptor code, by contrast, there are no constraints to the amount of data that can be protected by a single code, and processing complexity grows only linearly with source block size. As a result, a DF Raptor code allows full flexibility in choosing the protection period and protection amount for a streaming application without any negative consequences to performance or processing requirements.

Conclusion

Digital Fountain's DF Raptor erasure code has been designed and optimized to offer robust protection against packet loss without demanding excessive processing requirements. Digital Fountain's DF Raptor technology provides significant performance and practical advantages over comparable FEC technologies. Ultimately, a DF Raptor code helps ensure high-quality data delivery by protecting against packet loss with minimal overhead and processing requirements.

For streaming applications, constraints on the number of encoding symbols restrict the length of protection period that can be supported by a Reed-Solomon erasure code, thus limiting the robustness that can be achieved. Moreover, the high processing requirements associated with low code rates and long protection periods for Reed-Solomon erasure codes further limit their overall utility.

In comparison to Reed-Solomon erasure codes, a DF Raptor code is better suited for streaming applications. The relative merits of a DF Raptor erasure code versus Reed-Solomon erasure codes include:

- A DF Raptor code can adapt to the current level of packet loss as needed, while Reed-Solomon erasure codes typically require that the code rate be explicitly chosen for the expected level of maximum packet loss

- A DF Raptor code allows full flexibility in choosing the code parameters, while Reed-Solomon erasure codes impose constraints on the number of output symbols that directly affect their performance in protecting blocks of streaming data

- Processing requirements for a DF Raptor code grows linearly with the source block size, while Reed-Solomon erasure codes exhibit processing requirements that grow quadratically with source block size

For more information about Digital Fountain's advanced DF Raptor FEC technology, please visit www.DigitalFountain.com or contact info@DigitalFountain.com