ACCEPTED FROM OPEN CALL

Myths and Realities of Rateless Coding

Nicholas Bonello and Yuli Yang, King Abdullah University of Science and Technology (KAUST)
Sonia Aïssa, KAUST, INRS
Lajos Hanzo, University of Southampton

ABSTRACT

Fixed-rate and rateless channel codes are generally treated separately in the related research literature and so, a novice in the field inevitably gets the impression that these channel codes are unrelated. By contrast, in this treatise, we endeavor to further develop a link between the traditional fixed-rate codes and the recently developed rateless codes by delving into their underlying attributes. This joint treatment is beneficial for two principal reasons. First, it facilitates the task of researchers and practitioners, who might be familiar with fixed-rate codes and would like to jump-start their understanding of the recently developed concepts in the rateless reality. Second, it provides grounds for extending the use of the well-understood code-design tools — originally contrived for fixed-rate codes — to the realm of rateless codes. Indeed, these versatile tools proved to be vital in the design of diverse fixed-rate-coded communications systems, and thus our hope is that they will further elucidate the associated performance ramifications of the rateless coded schemes.

INTRODUCTION

Luby transform (LT) codes [1] are now deemed to be the first practical realization of rateless codes and were originally intended to be employed for recovering packets that are lost (erased) during transmission. Generally speaking, all rateless codes share a common trait of being capable of generating *any* number of output symbols from a given finite set of input (information) symbols. Metaphorically speaking, rateless codes are sometimes compared to an abundant water supply (fountain), which is able to supply an unlimited number of drops (i.e., redundant output symbols). Due to this reason, rateless codes are also referred to as "fountain" codes. LT codes also feature as components of some of the more contemporary rateless codes, such as the so-called Raptor codes.

The main focus of this article is on some of the pertinent issues that have resurfaced from the even more recent research efforts concerned with the deployment of rateless codes over time-varying noisy channels rather than over erasure channels. Against this backdrop, our forthcoming discussions will attempt to provide an answer to the following central questions:

- What are the similarities and differences between rateless codes and the previously proposed channel codes? What are the advantages and disadvantages of rateless codes with respect to the more traditional codes?
- In which circumstances would it be more preferable to employ a rateless code?

We will commence from the basic principles of rateless encoding and decoding and extend our analysis up to the current state-of-the-art code design techniques.

PRELIMINARIES

This section outlines the rudimentary principles of the encoding and decoding techniques of rateless coding, as originally presented in the context of LT codes [1] for transmission over erasure channels. Readers who are interested in a more developed treatment, would benefit from momentarily redirecting their attention to [2] for relevant pointers to the related literature.

ENCODING

Let us start by assuming a data message consisting of K input (source) symbols $\mathbf{v} = [v_1\ v_2 \cdots v_K]$, with each symbol containing an arbitrary number of bits. The terminology used in [1] refers to the original data message $\mathbf{v}$ as a "file."[1] For the sake of simplifying our discourse, let us consider a hypothetical example of a data message containing four two-bit symbols given by the vector $\mathbf{v}$ = [00 10 11 01]. Subsequently, the LT-encoded symbols $\mathbf{c} = [c_j]$,[2] $j = 1, \ldots, N$, is simply the modulo-2 sum (i.e., the eXclusive OR (XOR)) of d_c distinct input symbols, chosen uniformly at random. The actual d_c-value of each symbol to be encoded is then chosen from a pre-defined distribution $f(d_c)$ [1], which is also sometimes referred to as the output distribution. Figure 1 illustrates the original distribution by Luby in [1], which is known as the robust soliton distribution.

Let us further progress with our simple example and assume that the first d_c-value sampled from $f(d_c)$ is 2. Consequently, two input symbols will be chosen at random from $\mathbf{v}$. Let us for argument's sake assume that the two selected

[1] *As it will become more apparent in later sections, the rateless coding arena is barraged with equivalent or very similar nomenclatures, which will inevitably create some difficulty for a novice in the area. In fact, one of the aims of this treatise is to try to harmonize the related terminology developed.*

[2] *The encoded symbol sequence may also be referred to as a codeword. In the so-called non-systematic codes, the codeword is entirely constituted from parity-check bits/symbols.*

 0163-6804/11/$25.00

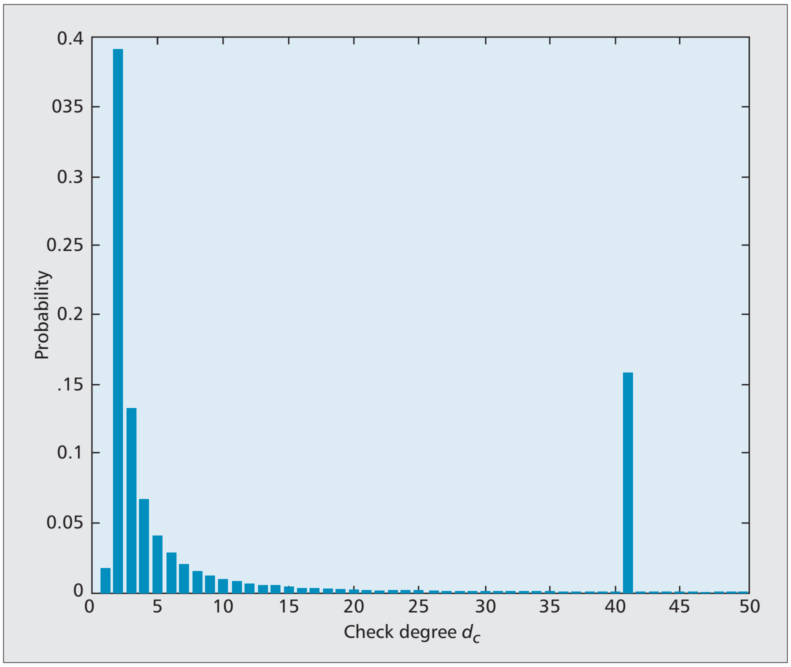

Figure 1. *A frequently used distribution $f(d_c)$, called the robust soliton distribution [1], which provides the probability of having a particular LT-encoded symbol c_j, $j = 1, \ldots, N$, generated from the XOR product of d_c pseudo-randomly selected source symbols in* **v**.

source symbols correspond to the first and second symbol in **v**. In such a case, c_1 will be calculated as $00 \otimes 10 = 10$, where "$\otimes$" denotes the XOR function. The encoding algorithm proceeds in the same manner, each time selecting a d_c-value sampled pseudo-randomly from $f(d_c)$ and then calculating the encoded symbol according to the XOR product of a d_c-number of pseudo-randomly selected source symbols. We also remark that some authors prefer to refer to the encoded symbols as the "LT-encoded packets" due to the natural applicability of LT codes to the Internet channel.

The connection between the input and output symbols can also be diagrammatically represented by means of a bipartite graph, as shown in Fig. 2. Using graph-related terminology, the input source symbol may be referred to as a *variable node*, while the LT-encoded symbol can be regarded as a *check node*. The number of edges emerging from each check node corresponds to the d_c-value sampled from $f(d_c)$ and for this reason, d_c is sometimes referred to as the *check degree*. Hereafter, we will interchangeably use the equivalent terminologies of input/output symbols, source/LT-encoded symbols, and variable/check nodes.

Decoding

Up to the current stage of our discourse, we are still considering erasure channels, in which encoded symbols of **c** are erased during the ensuing transmission. Simplistically speaking, the task of the decoder is that of recovering the erased symbols from the unerased ones. The decoding process as detailed in [1] commences by locating a degree-one input symbol; i.e., a symbol that is not XOR-combined with any other. The decoder will then modulo-2 add the value of this symbol to all the LT-encoded symbols relying on it and then removes the corresponding modulo-2 connections, where each connection is represented by an edge on the associated bipartite graph (Fig. 2). The decoding procedure continues in an iterative manner, each time starting from a degree-one symbol. If no degree-one symbol is present at any point during the decoding operation, the process will abruptly halt. However, a carefully designed degree distribution $f(d_c)$ guarantees that this does not occur more often than a certain pre-defined probability of decoding failure. This LT decoding process is diagrammatically explained in more detail in Fig. 4 of [3].

The Initial Challenges Faced

The initial challenges faced in the deployment of LT codes over time-varying noisy channels were very much focused on the decoding technique used. We recall that the aforementioned decoder outlined earlier can only *recover* lost symbols. Compared to erasure channels, however, noisy channels raise different obstacles to overcome, since the decoder must gain a superior capability of *correcting* the symbols corrupted by the channel. In fact, it soon became evident that if a single corrupted bit (in a symbol) was left participant in the aforesaid decoding process, this would inevitably corrupt all other symbols connected to it. This first step toward attaining a solution to this problem was the formalization of a concept called "wireless erasure" [4]. The employed technique was conceptually simple; a cyclic redundancy check (CRC) sequence was appended to every block containing an (arbitrary) number of LT-encoded symbols and if their CRC fails, then this particular block of symbols was declared to be erased and forbidden from participating in the ensuing decoding process. In such a manner, a noisy channel was effectively being treated like a block erasure channel. Another somewhat related technique exploited the log-likelihood ratio[3] (LLR) as a measure of confidence in the alleged correctness of every received bit in each symbol. Subsequently, each symbol containing bits with LLR values smaller than a predetermined threshold were prohibited from participating in the decoding process [4]. However, both techniques had their own limitations; indeed, both would declare a potentially large number of correct bits as being erased and so, the associated rateless code fails to operate very close to the ultimate capacity limits.

Two noteworthy contributions were then put forth by Jenkac *et al*. [4] and Palanki and Yedidia [5]. The former suggested a superior decoding strategy for LT-coded transmissions over noisy channels, by allowing the exchange of LLRs (also known as "soft decoding") between the source and LT-encoded symbols, in a similar manner to that employed in the decoding of low-density parity-check (LDPC) codes and repeat-accumulate (RA) codes. From this point onward, the encoding and decoding operations for rateless codes were executed on a bit-by-bit basis, rather than on a symbol-by-symbol basis, since the associated computational decoding complexity of the

[3] *The LLR of a decoded bit is defined by the logarithm of the ratio of two probabilities, i.e., the probability of the bit taking values of '+1' and '–1.' The sign of the LLR indicates the polarity of the bit while the magnitude of the LLR gives an indication of the probability of making a correct decision.*

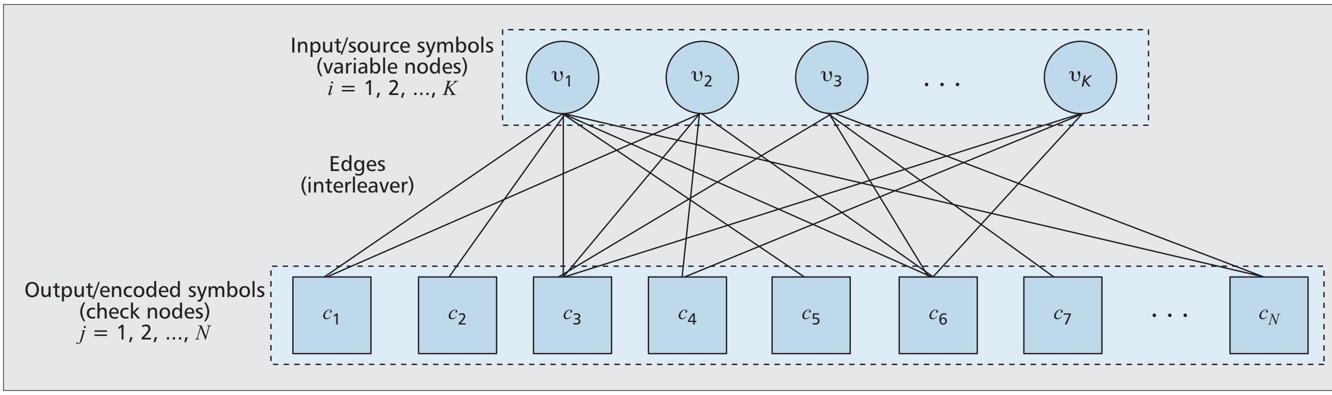

Figure 2. *A bipartite-graph-based description of an LT code showing the source symbols (variable nodes) and the LT-encoded symbols (check nodes). The symbols can be of an arbitrary size. For instance, c_4 is calculated by the modulo-2 sum (XOR product) of v_2 and v_K in the specific example portrayed in this bipartite graph.*

latter would be prohibitively complex for symbols of large sizes, as is the case with LT codes.[4]

Despite the unmistakable performance improvement offered by the more sophisticated soft decoding algorithm, LT codes still failed to exhibit an acceptable performance over noisy channels. In fact, the bit error ratio (BER) and block error ratio (BLER) results illustrated in Fig. 2 of [5] illustrate high error floors when transmitting over binary symmetric or binary-input additive white Gaussian noise (BIAWGN) channels. The result presented in [5] implicitly suggested that LT codes may have to be concatenated with other forward error correction (FEC) schemes in order to improve their exhibited performance. Indeed, a substantial amount of papers in the literature of this era advocate a number of diverse LT-concatenated schemes using components as simple as convolutional codes to more complex turbo codes and LDPC codes (please refer to [2] and references therein).[5] The latter — commonly known as Raptor codes [6] — have probably received the most attention in the available literature. Similarly to their predecessors, Raptor codes were also originally contrived for transmissions over erasure channels. However, in contrast to LT codes, Raptor codes were capable of maintaining a good performance over noisy channels (please refer to Fig. 2 of [5]). For this reason, Raptor codes started to slowly overshadow LT codes, thus becoming the first choice for rateless coding over noisy channels.

Classification of Channel Codes

Generally speaking, channel codes can be categorized into three classes: the class of fixed-rate codes, the class of rateless codes, and the in-between class of finite discrete-rate codes. In this section, we will elaborate further on the similarities and differences between the three classes, which will also be summarized in Table 1.

Let us commence with the well-understood fixed-rate codes. As their name in itself implies, codes in this category encode every information bit using the same amount of redundancy. A fixed-rate code having a rate R_x can be carefully designed to attain a performance that is close to the capacity target $C(\psi_x)$ at a specific channel signal-to-noise ratio (SNR) value of ψ_x dB, for which it was originally contrived. However, such codes are unsuitable for employment over time-varying channels because if the encountered SNR is actually higher than ψ_x dB, the fixed-rate code essentially becomes an inefficient channel code — albeit its good performance at ψ_x dB — since the code incorporates more redundancy than the actual channel conditions require. On the other hand, if the channel SNR encountered becomes lower than the SNR value of ψ_x dB, then the channel code fails to supply sufficient redundancy to cope with the channel conditions encountered.

In this light, it becomes evident that adaptation plays an important role in coping with channels of time-varying quality. A feasible but not so effective solution is to employ several dedicated fixed-rate channel codes having different code rates, which are selectively switched on according to the channel quality. However, this scheme requires:

- An encoder and decoder pair for each channel code implemented, thus its associated complexity may become prohibitively complex.
- Channel state information (CSI) feedback from the receiver.

A more pragmatic approach is to modify the code-rate of a fixed-rate channel code by using *code puncturing* or *code extension* techniques. Code puncturing involves removing some of the codeword bits and thus creating a code having a rate that is higher than the original code of rate R_x (called the *mother code*), while code extension adds more parity bits and thus reduces the code-rate. Such techniques are typically applied in the context of the so-called type-II hybrid automatic repeat-request (HARQ) schemes. In such systems, the transmitter continues to send additional parity-bits of a codeword until a positive acknowledgment (ACK) is received or all redundancy available for the current codeword are sent.

In the available literature, such schemes are typically referred to by the somewhat related terminology of "variable-rate," "rate-compatible," "adaptive-rate" or "incremental redundancy" schemes. However, our concern with such terminology is that it can be equally suitable for describing rateless codes. In an attempt to bring forth a more distinguishable terminology, we prefer to employ the nomenclature of "finite discrete-rate" codes, because of their capability to

[4] *Consequently, we will interchangeably use the terminology of "bits" and "symbols" in all the forthcoming discourse.*

[5] *However, the available literature does not provide explicit justifications for (a) the inability of LT codes to achieve good performance over noisy channels, and (b) the capability of aforementioned concatenated rateless schemes to succeed where the LT codes fail. Section VI of this treatise contributes (in part) to fill the small gaps in this regard.*

Attributes	Fixed-Rate	Finite Discrete-Rate	Rateless
Rate	Does not change during the transmission of a codeword	Varies between a finite set of predetermined code-rates	Can potentially realize codes having any code-rate
Degree Distribution	Does not change	May vary between rates	May vary between certain instantaneous rates
Code Description	Remains the same for every transmitted codeword	Remains the same for every codeword[2] transmitted with a specific code-rate	Varies for every transmitted codeword
Feedback Channel	Not required	Required for receiving ACKs from the receiver	Required for receiving ACKs from the receiver
Encoder Complexity	Depends on the specific code chosen	Comparable to the respective fixed-rate code	Their computational complexity is comparable to their fixed-rate counterpart. Their implementational complexity is simpler since there is no need to store the code description
Decoder Complexity	Depends on the specific code chosen	Their implementation complexity is comparable to their respective fixed-rate code. Their computational complexity may be higher due to the multiple decoding attempts	Their implementational complexity is comparable to their fixed-rate counterpart. Their computational complexity may be higher due to the multiple decoding attempts
Code Design[1]	Possible	Possible	Very limited due to the dynamic nature of the code

[1] Here we are referring to the possibility of using code design techniques such as girth and/or cycle conditioning (please refer to [2] and references therein).

[2] Additional details on this specific attribute are provided in the subsection "Underlying Assumptions" in the section "Further Pertinent Issues."

Table 1. *A comparison of various attributes manifested by fixed-rate, finite discrete-rate and rateless codes.*

realize codes from a *finite* discrete set of code-rates. On the other hand, the inherent flexibility of rateless codes allows them to potentially realize codes having *any* code-rate from a *countably infinite* discrete set of rates. Indeed, it can easily be inferred from our preliminary discussions earlier that there is no limit on the possible number N of encoded symbols/bits that a rateless code can produce. However, defining the rate of a rateless code requires some further thought. In [7], we differentiate between what we refer to as the *instantaneous rate* and the *effective rate*. The instantaneous rate is determined by the ratio of the number of information bits K to the total number of bits transmitted at *a specific instant*. On the other hand, the effective rate signifies the rate realized at the specific point when the codeword has been successfully received at the receiver. This instantaneous versus effective terminology may also be extended to other parameters of the rateless codes, e.g, to the block length.[6] The formalization of this terminology constitutes the first step toward bridging the link between rateless and the traditional fixed-rate codes, which constitutes the topic of our next section.

Rateless Paradigms

LT codes can be regarded as the rateless counterpart of the fixed-rate, irregular,[7] non-systematic[8] low-density generator matrix (LDGM)-based codes [8]. In this regard, the encoding process described earlier can be more conveniently described by means of a time-variant generator matrix $\mathbf{G}$ having K rows and N columns. The parameter N represents the total number of encoded symbols/bits; i.e., the effective block length of the code,[9] where we have $N > K$. Let v_i and $G_{i,j}$ denote the ith symbol/bit of the information sequence $\mathbf{v}$ and the element in the ith row and jth column of $\mathbf{G}$, respectively, where $i = 1, \ldots, K$ and $j = 1, \ldots, N$. The encoded symbol c_j is then calculated by the modulo-2 sum of all the v_i's whose corresponding $G_{i,j}$ is non-zero; i.e., $c_j = \sum_{i=1}^{K} v_i\, G_{i,j}$.

If we reconsider the simple example introduced earlier, where the information bit sequence was equal to $\mathbf{v}$ = [00 10 11 01], and assume a generator matrix $\mathbf{G}$ given by

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{1}$$

then the LT-encoded symbols $\mathbf{c} = [c_1 \cdots c_N]$ would be equal to [10 00 00 11 00 00 11 11]. The generator matrix of an LDGM/LT code is said to have a low density since it contains more zero than non-zero elements. We also remark that there is a one-to-one relationship between $\mathbf{G}$ and the bipartite graph of the associated code; the

[6] *This terminology is also suitable for the finite discrete-rate codes, in light of which may be viewed to realize codes having a predetermined finite set of instantaneous code-rates.*

[7] *The word "irregular" indicates that the number of edges emerging from each variable (or check) node in the underlying bipartite graph (Fig. 2) is kept constant.*

[8] *The word "non-systematic" indicates that the K information bits do not form part of the transmitted codeword.*

[9] *In light of the earlier discussions, the effective block length can be defined as the total number of parity-check symbols/bits that have to be transmitted for the receiver to correctly recover the original K information bits.*

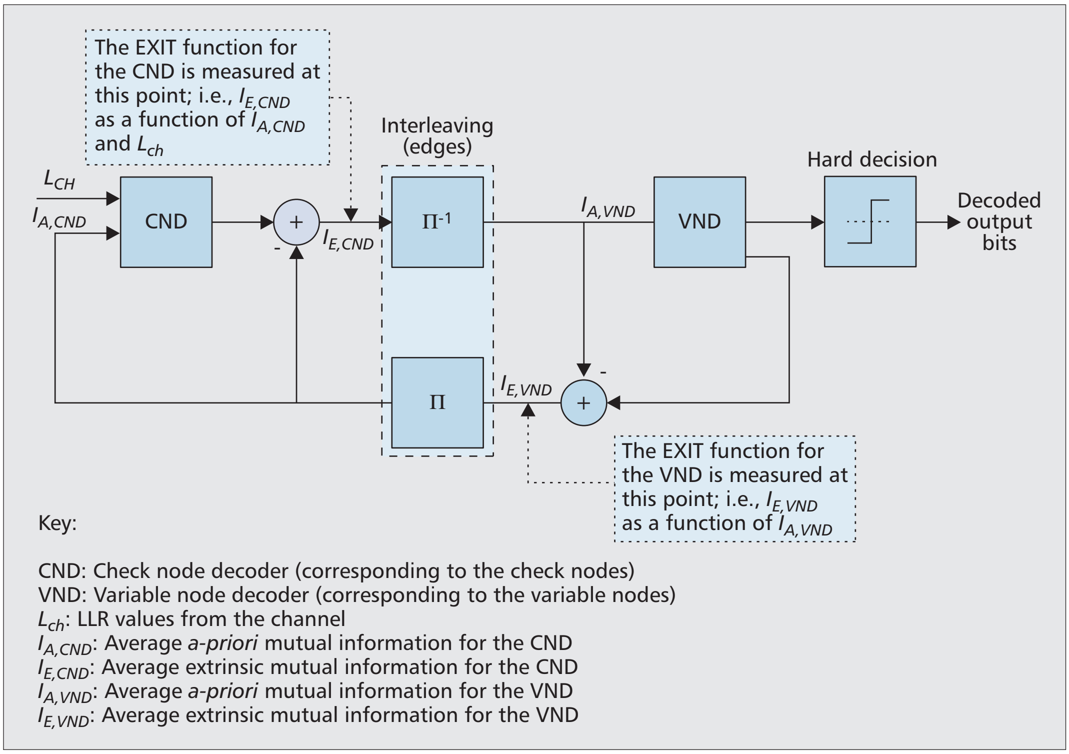

Figure 3. *Decoder structure of an LT code.*

One must emphasize that while the majority of rateless codes are mirrored by a code in the fixed-rate domain, the reverse is not necessarily true. Two noteworthy examples of this phenomenon are turbo codes and LDPC codes

rows, columns and non-zero elements of **G** correspond to the variable nodes, check nodes, and interconnecting edges of the graph. For example, Fig. 2 portrays the graph identified by the **G** formulated in Eq. 1, assuming $K = 4$ and $N = 8$.

The differences between traditional (i.e., fixed-rate) LDGM codes and LT codes can be summarized in the following two points:

- The generator matrix of LT codes is calculated online during the encoding process, whereas that of traditional LDGM codes is time-invariant and thus can be hardwired.
- The N-number of columns of **G** is not fixed and thus can change for every information sequence to be transmitted. Every LT-encoded symbol c_j transmitted involves adding another column to **G**, which describes the edge interconnections between c_j and the respective v_i's. The value of N will actually depend on the erasure probability if the channel considered is the erasure channel or on the value of the SNR for transmissions over noisy channels.

Subsequently, we can further extend the analogy between other rateless codes and their fixed-rate counterpart. For instance, the encoding process of the aforementioned Raptor codes [6] starts by first encoding the information sequence **v** by a typically high-rate (fixed-rate) LDPC code, and then re-encoding the symbols/bits at the output of the LDPC code by an LT code. Consequently, Raptor codes may be viewed as the serial concatenation of a fixed-rate LDPC code with a rateless LDGM code. Along similar lines, the encoding process of the reconfigurable rateless codes of [7] also involves two encoder stages: first, we encode **v** using an LT code as described earlier, and then re-encode the symbols/bits at the output of the LT code with a unity-rate code, which is referred in parlance as "accumulator." Hence, reconfigurable rateless codes may be deemed to be rateless LDGM codes concatenated with an accumulator; i.e., an instance of rateless RA codes. However, one must emphasize that while the majority of rateless codes are mirrored by a code in the fixed-rate domain, the reverse is not necessarily true. Two noteworthy examples of this phenomenon are turbo codes and LDPC codes, both of which do not possess a rateless similitude due to their lack of flexibility in supporting the rateless encoding process, akin to that outlined earlier.

Analysis of Rateless Codes Through EXIT Charts

By regarding rateless codes as instances of other fixed-rate codes, we can proceed to employ the well-understood code-design tools of the latter in order to explain the associated performance ramifications of the former codes. Our forthcoming discussion will still be very much focused on LT codes. After all, LT codes are the constituent codes of the majority of the rateless codes proposed in the current literature.

Let us first start by explaining the structure of an LT decoder. This is illustrated in Fig. 3 and it closely follows the construction of the associated bipartite graph (Fig. 2), which is in turn related to the generator matrix **G** of the code. We observe from Fig. 3 that the LT decoder contains two serially-concatenated constituent decoders: a check node decoder (CND) and a variable node decoder (VND), which correspond to the check and variable nodes in the graph. At each iteration, the CND and VND will exchange soft information (in terms of the previously described LLR values) about each bit through the separating edge interleaver, which corresponds to the edges in the associated bipartite graph. In this context, we generally distinguish between three different types of information: the a-priori, the a-posteriori, and the extrinsic information. The a-priori information represents that information about a bit available before the start of the decoding; i.e., the information available at the input of a decoder that accrues from the other constituent decoder. The information produced by the decoder is then called the a-posteriori information. Subsequently, the extrinsic information is the a-posteriori information excluding the a-priori information. Thus we can observe from Fig. 3 that the CND receives both channel output values as well as the a-priori LLRs from the VND and then converts them to a-posteriori LLRs. The VND receives the de-interleaved extrinsic information from the CND.

Elaborating slightly further, the CND and VND decoders can be mathematically described by the so-called extrinsic information transfer (EXIT) function.[10] The LT decoder can therefore be represented by two EXIT functions, one for each constituent code, which will hereby be denoted by $I_{\text{E,CND}}$ and $I_{\text{E,VND}}$. These functions describe the extrinsic information output of the CND or VND as a function of their a-priori information input $I_{\text{A,CND}}$ or $I_{\text{A,VND}}$. Plotting $I_{\text{E,CND}}$ versus $I_{\text{A,CND}}$ and $I_{\text{E,VND}}$ versus $I_{\text{A,VND}}$ will produce two curves, called the *EXIT curves*, and the resulting plot is referred to as the *EXIT chart*. The latter has proved to be a very useful tool for designing FEC codes, because it provides a reasonably accurate prediction of their performance, without performing the actual bit-by-bit decoding using the time consuming Monte Carlo simulations. Assuming this EXIT chart-based framework, an LT code will exhibit a good performance if:

- Both the CND and VND EXIT curves reach the uppermost right-hand side corner (i.e., the (1, 1) point) of the EXIT chart.
- The CND EXIT curve always stays above the VND EXIT curve and never intersects; i.e., there is an open tunnel between the two EXIT curves.
- The two EXIT curves match each other as accurately as possible to minimize the area between them. If the first two requirements are not met, then the resultant BER/BLER performance will exhibit high error floors.

The third requirement is indispensable for attaining a near-capacity performance. One would of course expect that at least some of these requirements will not be satisfied in the associated EXIT chart of an LT code transmitting over noisy channels, since we have already alluded earlier to the fact that LT codes fail to achieve a good performance over such channels. In this light, the EXIT chart is merely being used as a diagnostic plot in order to identify where the deficiencies in the code are and to indicate the potential remedies.

Figure 4a portrays the EXIT chart of an LT code for transmission over the BIAWGN channel at energy-per-bit to the noise power spectral density ratio of $E_b/N_0 = 5$ dB and employing the aforementioned robust soliton distribution. There are three main points that can be deduced from the EXIT chart of Fig. 4a. First, it can be observed that despite having an open tunnel between the two EXIT curves, the EXIT function of the CND does not reach the point (1, 1) in the EXIT chart, where decoding convergence to an infinitesimally low BER may be expected. The second issue that requires further consideration is the low starting value of the CND's EXIT curve. The higher the starting value, the higher the probability that the decoding process is triggered toward the point (1, 1). The low starting value of the CND curve is due to the low number of degree-one check nodes in the check node degree distribution $f(d_c)$, which will inevitably force the LT decoder to expend a substantial number of iterations in order to attain minute improvements in the quality of its extrinsic information at the beginning of the decoding process. Third, it can also be observed from Fig. 4a that the two EXIT curves do not match very closely, which indicates that a near-capacity performance cannot be attained. In this context, we remark that the shape of the EXIT curve very much depends on the underlying degree distribution. Therefore, the second and third aforementioned issues can be tackled by carefully redesigning appropriate distributions, both for the check and variable node[11] degrees, for rateless coded transmissions over noisy channels. Recent work focusing on these aspects include [7, 10, 11]. On the other hand, the first-mentioned difficulty[12] can be circumvented by serially concatenating the LT code with another code, and specific realizations of such schemes have already been mentioned earlier. For the sake of comparison, Fig. 4b displays the EXIT chart for a rateless code comprised of the serial concatenation of an LT code and a (unity-rate) accumulator, transmitting over BIAWGN channels with $E_b/N_0 = 5$ dB and having an instantaneous/effective code-rate of 0.93. Besides the considerable increase in the achievable throughput at the same E_b/N_0 value of 5 dB, one can also observe that concatenation as well as the appropriate design of the distributions for each constituent code [7, 11] succeed in solving all the deficiencies manifested in Fig. 4a.

Further Pertinent Issues

In this section, we will outline a number of issues of relevance to any rateless code, some of which have not yet been thoroughly investigated in the available literature.

Underlying Assumptions

A fundamental assumption of any rateless coding scheme is that the receiver has exact knowledge of the degree d_c of each received bit and its specific modulo-2 connection(s) to the original information bits. Indeed, it is absolutely vital that both the transmitter and the receiver share the same code description because if this is not the case, then the decoder at the receiver would

[10] *For the sake of completeness, we would like to add that the EXIT function describes the so-called average extrinsic mutual information of the respective decoder. Additional details are beyond the scope of this article, and so we refer the interested reader to [9] and the references in [2] for further details.*

[11] *The variable node distribution determines the shape of the VND's EXIT curve and corresponds to the distribution of the non-zero elements across the rows of the code's generator matrix **G**. Although this distribution is not explicitly specified for LT codes, it can be reasonably approximated by the Poisson distribution. Consequently, there will be some rows in the LT code generator matrix with a low number of non-zero elements, regardless of the length of the code N. By analogy to fixed-rate LDGM codes, the minimum distance for LT codes will be at most r, where r represents the smallest number of non-zero elements across the K rows of **G**. In this regard, it may be argued that LT codes also possess poor distance properties.*

[12] *We remark that these deficiencies of LT codes are also shared by their fixed-rate counterparts; i.e., the non-systematic LDGM codes. These codes are known to exhibit high error floors (see for example [8]). The BER/BLER performance of LDGM codes is usually improved by using serially concatenated structures created by combining them, for example, with another LDGM code (please refer to [8, 12], among others.)*

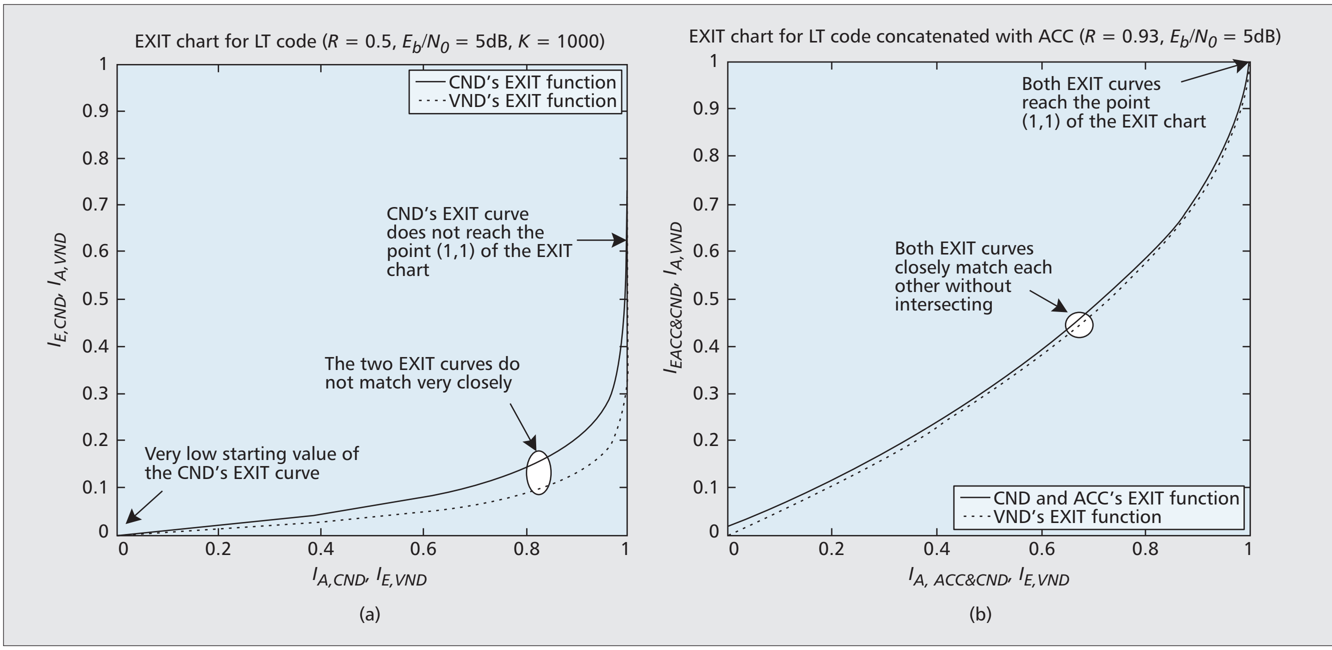

Figure 4. *For a rateless code, the EXIT chart may be deemed to be either a "snapshot" of the performance exhibited at an instantaneous code-rate, or else the actual performance attained at the effective code-rate. The terminology of instantaneous versus effective code-rate is discussed in the section on "Analysis of Rateless Codes Through EXIT Charts." Figure (a) illustrates the EXIT chart for an LT code [1], transmitting over BIAWGN channels with* $E_b/N_0 = 5$ *dB and assuming an instantaneous/effective code-rate of 0.50. Figure (b) displays the EXIT chart for a rateless code consisting of the serial concatenation of an LT code and a (unity-rate) accumulator. The parameters* $I_{A.ACC\&CND}$ *and* $I_{E.ACC\&CND}$ *respectively denote the average a-priori and extrinsic information of the combined accumulator (ACC) and CND.*

be operating on a different code than the one used by the encoder at the transmitter.[13] This difficulty is circumvented either by assuming that:

1 An additional header is appended with every block of (an arbitrary number of) bits to describe the required information about the check degrees and the specific modulo-2 connection(s), or else by assuming that
2 The transmitter and the receiver have synchronized clocks used for the seed of their pseudo-random number generators and so, the degree d_c and the specific modulo-2 connections selected by both the transmitter and the receiver will be identical [3].

Only the second assumption is applicable to support the employment of rateless codes over noisy channels since an appended header (as suggested in the first solution) is of course liable to become corrupted during transmission. To the best of our knowledge, there exists no detailed analyses of the effect of the extra overhead on the exhibited throughput performance in solution 1 and on the vulnerability of rateless coding schemes in case of violation of assumption 2.

Degree Distribution

As briefly mentioned earlier, rateless codes employ the degree distribution $f(d_c)$ for coining the degree d_c for each transmitted bit. However, for conventional rateless codes such as in [1, 6], $f(d_c)$ is fixed before transmission. Consequently, one may argue that such rateless codes can only alter the number of bits transmitted (i.e., the code-rate) in order to counteract the variations of the channel conditions encountered. However, it was shown in [13] that a degree distribution designed for rateless coded transmissions over time-varying noisy channels will inevitably depend on the channel characteristics and so, a single fixed degree distribution can never be the optimal distribution for all the diverse channel conditions encountered. Reconfigurable rateless codes were subsequently proposed in [7]. These codes exploit better the inherent dynamic characteristics of rateless codes by not only varying their code-rate (like other conventional codes), but also by adaptively modifying their degree distribution according to the prevalent channel conditions. Figure 5 illustrates a comparison of their achievable throughput with that exhibited by Raptor codes [6], punctured regular and optimized irregular LDPC codes. The discrete-input continuous-output memoryless channel's (DCMC) capacity curve is also shown. The Raptor code and the punctured LDPC benchmarker codes follow the same designs presented in [14]. The decoder employs the well-known sum-product algorithm, limited to a maximum of 200 iterations. The number of information bits used for all the simulated schemes was set to 9,500 bits.

Feedback Requirements

Rateless (as well as the finite discrete-rate) codes are capable of adapting to time-invariant channel conditions without requiring explicit CSI feedback from the receiver. However, we emphasize that this does not imply that rateless codes can entirely dispose of the feedback channel; on the contrary, there is still the necessity of having a *reliable* feedback channel for the receiver to

[13] We remark that this is not of an issue in fixed-rate and finite discrete-rate coding schemes, since the code description in these two channel code families is time-invariant and thus can be stored in

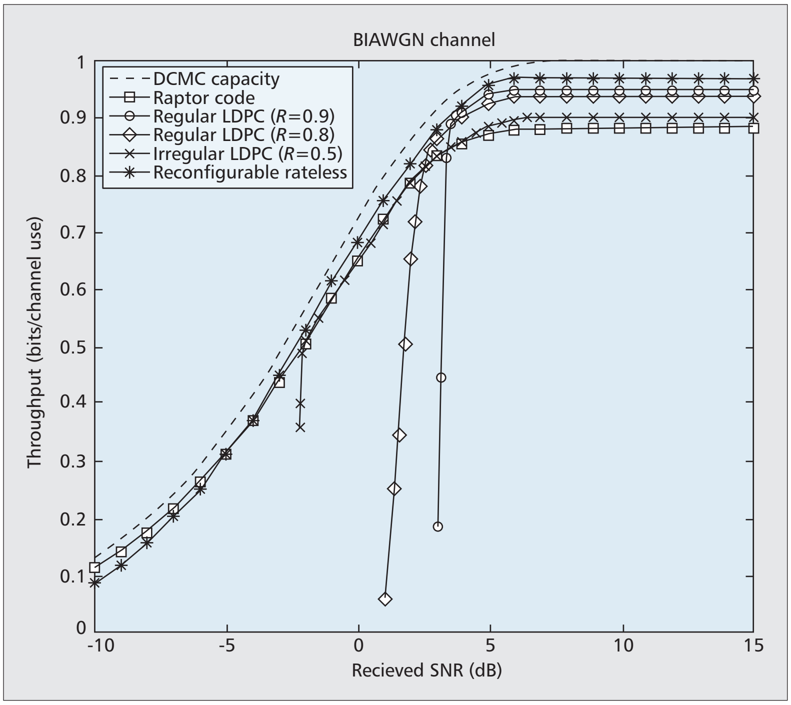

Figure 5. *Average throughput (in bits/channel use) performance for transmission over BIAWGN channels versus SNR (in dB) using reconfigurable rateless codes [7], Raptor codes [6], incremental-redundancy-based HARQ schemes employing punctured regular LDPC codes having code-rates of R = 0.8 and 0.9 and an optimized punctured half-rate irregular LDPC code.*

acknowledge the correct recovery of the data and thus allow for the next codeword's transmission to start. Additionally, it is worth underlining that the aforementioned reconfigurable rateless codes [7] can be appropriately designed to adapt their distribution based on the *absence* of the receiver's ACKs, and so their feedback requirements are similar to their conventional counterparts [1, 6]. The available research literature would benefit from detailed investigations on the achievable performance of rateless coded schemes employing practical, non-idealized feedback channels.

Code Optimization

The BER/BLER performance exhibited by the so-called codes-on-graphs[14] is affected by a number of graph-theoretical properties such as the underlying girth as well as the presence of stopping sets and trapping sets, among others. Consequently, recent research efforts focused their attention on optimizing these fixed-rate codes with respect to their underlying graphical attributes. However, despite the fact that the majority of rateless codes also possess a graphical description (Fig. 2), the extension of these code-optimization techniques to the rateless domain is difficult if not inapplicable, because as we have seen previously, the parity-check connections between the information and parity bits of a rateless code are time-variant and determined "on-the-fly." With the exception of the very recent work of [15], the understanding of the aforementioned graphical attributes on the exhibited performance of rateless codes is still very much in its infancy.

[14] As the name in itself implies, codes-on-graphs encompass a relatively broad category of FEC codes that can be described by means of graphs; i.e., LDPC codes and their relatives.

Conclusion

We began this article by outlining the encoding and decoding techniques employed by rateless codes, as originally presented by Luby for LT codes [1], when transmitting over erasure-inflicted channels. Following this preliminary foundation, we have explored some of the related issues that transpired following the initial efforts of deploying rateless codes over time-varying noisy channels. Subsequently, we have attempted to provide a broader picture to our readers by discussing the similarities as well as the differences of rateless codes with respect to their more traditional channel coding counterparts of fixed-rate and finite discrete-rate codes. The most prominent attributes of each channel coding family were also summarized in Table 1. Following this, we discussed several rateless paradigms by presenting some rateless codes' realizations as instances of traditional fixed-rate codes. These discussions provided the necessary underlying framework for using EXIT charts, which served as a good eye opener to additional insights on their exhibited performance over noisy channels. Finally, we outlined a number of concomitant topics, some of which have not yet received enough attention in the related literature. We expect that some of the ideas presented in this article will be in the future extended for the design of rateless codes for transmission over channels that are characterized by noise as well as erasures.

References

[1] M. G. Luby, "LT Codes," *Proc. 43rd Annual IEEE Symp. Foundations of Computer Science*, Nov. 16–19, 2002, pp. 271–80.

[2] N. Bonello, S. Chen, and L. Hanzo, "LDPC Codes and Their Rateless Relatives," accepted in the *IEEE Commun. Surveys and Tutorials*, available online from http://eprints.ecs.soton.ac.uk/17394/1/Survey LDPC.pdf.

[3] D. J. C. MacKay, "Fountain Codes," *IEE Proc. Commun.*, vol. 152, Dec. 9, 2005, pp. 1062–68.

[4] H. Jenkac *et al.*, "Soft Decoding of LT-Codes for Wireless Broadcast," *Proc. IST Mobile and Wireless Summit*, June 19–23, 2005.

[5] R. Palanki and J. S. Yedidia, "Rateless Codes on Noisy Channels," *Proc. IEEE Int'l. Symp. Info. Theory*, (Chicago, IL, USA), June 27–July 2, 2004, p. 37.

[6] M. A. Shokrollahi, "Raptor Codes," *IEEE Trans. Info. Theory,* vol. 52, June 2006, pp. 2551–67.

[7] N. Bonello *et al.*, "Reconfigurable Rateless Codes," *IEEE Trans. Wireless Commun.*, vol. 8, Nov. 2009, pp. 5592–600.

[8] J. Garcia-Frias and W. Zhong, "Approaching Shannon Performance by Iterative Decoding of Linear Codes with Low-Density Generator Matrix," *IEEE Commun. Letters*, vol. 7, June 2003, pp. 266–68.

[9] S. ten Brink, "Convergence of Iterative Decoding," *IEE Electronics Letters*, vol. 35, May 13, 1999, pp. 806–08.

[10] A. Venkiah, C. Poulliat, and D. Declercq, "Jointly Decoded Raptor Codes: Analysis and Design for the BIAWGN Channel," *EURASIP J. Wireless Commun. and Networking*, vol. 2009, 2009.

[11] N. Bonello *et al.*, "Generalized MIMO Transmit Preprocessing Using Pilot Symbol Assisted Rateless Codes," *IEEE Trans. Wireless Commun.*, vol. 9, Feb. 2010, pp. 754–63.

[12] M. Gonzalez-Lopez *et al.*, "Serially-Concatenated Low-Density Generator Matrix (SCLDGM) Codes for Transmission over AWGN and Rayleigh Fading Channels," *IEEE Trans. Wireless Commun.*, vol. 6, Aug. 2007, pp. 2753–58.

[13] O. Etesami and M. A. Shokrollahi, "Raptor Codes on Binary Memoryless Symmetric Channels," *IEEE Trans. Info. Theory*, vol. 52, May 2006, pp. 2033–51.

[14] E. Soljanin, N. Varnica, and P. Whiting, "Punctured vs. Rateless Codes for Hybrid ARQ," *Proc. IEEE Info. Theory Wksp.*, (Punta del Este, Uruguay), 2006, pp. 155–59.

[15] V. L. Orozco, “Trapping Sets in Fountain Codes Over Noisy Channels,” M.Sc. thesis, Queen’s University, Canada, 2009.

Biographies

NICHOLAS BONELLO (nicholas.bonello@kaust.edu.sa) received the B.Eng. (Hons) degree in Electrical Engineering from the University of Malta in 2004, the M.Sc. degree in Radio Frequency Communications Systems from University of Southampton, U.K., in 2006 and the Ph.D. degree in Wireless Communications from the University of Southampton, U.K, in 2009. He is currently a Postdoctoral Fellow at King Abdullah University of Science and Technology (KAUST), Kingdom of Saudi Arabia. His primary research interests include fixed-rate as well as rateless error correction techniques, signal processing and statistics.

YULI YANG [M] (yuli.yang@kaust.edu.sa) received her Ph.D. degree in Electrical Engineering from Peking University, China, in 2007. She is a Postdoctoral Fellow at King Abdullah University of Science and Technology (KAUST), KSA. Her industry experience includes internship with Huawei Technologies from 2006 to 2007, and research scientist with Alcatel-Lucent Shanghai Bell Labs from 2007 to 2009. Her research interests include information theory of wireless channels and networks, multi-antenna systems, cooperative communications and cross-layer wireless network design.

SONIA AISSA [SM] (sonia.aissa@ieee.org) received her Ph.D. degree in ECE from McGill University, Canada, in 1998. Since then, she has been with Institut National de la Recherche Scientifique (INRS), University of Quebec, Canada, where she is a Professor. From 1996 to 1997, she was a Researcher at Kyoto University and NTT, Japan. From 1998 to 2000, she was a Research Associate at INRS. From 2000 to 2002, she was a Principal Investigator in the major program of personal and mobile communications of the Canadian Institute for Telecommunications Research. From 2004 to 2007, she was an Adjunct Professor with Concordia University, Canada. In 2006, she was Visiting Invited Professor at Kyoto University. Her research activities are in the area of wireless communications and networking. She was the Founding Chair of the Montreal Chapter IEEE Women in Engineering Society in 2004–2007, acted or is currently acting as Technical Program Leading Chair or Cochair for the Wireless Communications Symposium of IEEE ICC in 2006, 2009, 2011 and 2012, and as PHY/MAC Program Chair for IEEE WCNC 2007. She is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON COMMUNICATIONS, and the *IEEE Communications Magazine*, and Associate Editor of the *Wiley Security and Communication Networks Journal*. Awards to her credit include the Quebec Government FQRNT Strategic Fellowship in 2001–2006; the INRS Performance Award in 2004 for outstanding achievements in research, teaching and service; and the Technical Community Service Award from the FQRNT center of excellence SYTACom in 2007. She is also co-recipient of Best Paper Awards from the IEEE Communications Society and recipient of NSERC Discovery Accelerator Supplement Award.

LAJOS HANZO [F] (lh@ecs.soton.ac.uk) received his degree in electronics in 1976 and his doctorate in 1983. During his 31-year career in telecommunications he has held various research and academic posts in Hungary, Germany and the UK. Since 1986 he has been with the School of Electronics and Computer Science, University of Southampton, UK, where he holds the chair in telecommunications. He has co-authored 17 books on mobile radio communications totaling in excess of 10000 pages, published in excess of 800 research papers, acted as TPC Chair of IEEE conferences, presented keynote lectures and been awarded a number of distinctions. Currently he is directing an academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council (EPSRC) UK, the European IST Programme and the Mobile Virtual Centre of Excellence (VCE), UK. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is also an IEEE Distinguished Lecturer as well as a Governor of both the IEEE ComSoc and the VTS. He is the acting Editor-in-Chief of the IEEE Press. For further information on research in progress and associated publications please refer to http://www-mobile.ecs.soton.ac.uk.