ACCELERCOMM
SUPERCHARGED WIRELESS

The 5G Channel Code Contenders

Robert G. Maunder
CTO AccelerComm

Abstract

Channel coding is a vital but complex component of cellular communication systems, which is used for correcting the communication errors that are caused by noise, interference and poor signal strength. The turbo code was selected to provide channel coding in 3G and 4G cellular systems, but the 3GPP standardization group is currently debating whether it should be replaced by the Low Density Parity Check (LDPC) or polar code in 5G. In this white paper, we dispel two myths that have been the genesis of this debate, namely that turbo codes cannot support high throughputs and that they have higher complexities than LDPC and polar codes. Indeed, AccelerComm has already demonstrated that turbo codes can achieve decoded throughputs exceeding the 5G target of 20 Gbps. Furthermore, this white paper shows that the overall implementation complexity of turbo, LDPC and polar codes depends on more than just their computation complexity. Since turbo codes have the benefits of low interconnect complexity and inherent flexibility, we show that they facilitate high-throughput flexible channel coding at lower implementation complexities than LDPC and polar codes. Furthermore, turbo codes offer the additional benefit of backwards compatibility to 3G and 4G, which offers the cellular communications industry some very significant cost savings. It is for these reasons that AccelerComm is promoting the turbo code for 5G.

I. What is channel coding?

In cellular communication systems, wireless transmission is used to convey data between handsets and basestations, where the latter act as gatekeepers to the Internet and telephone networks. However, the received data typically differs to the transmitted data, owing to communication errors caused by noise, interference or poor signal strength. In order to correct these communication errors, cellular communication systems use channel codes. More specifically, a channel encoder is used in the transmitter (be it the handset or the basestation) to convert each so-called data block comprising K data bits into a longer encoded block comprising $N > K$ encoded bits, which are transmitted. In the receiver, the additional $(N - K)$ encoded bits provide the channel decoder with information that allows it to detect and correct communication errors within the original K data bits.

If the noise, interference or poor signal strength is particularly severe, then a low coding rate $R = K/N$ will be required for the channel decoder to successfully detect and correct all transmission errors. However, a low coding rate implies the transmission of a high number N of encoded bits, which consume transmission energy and bandwidth. Therefore, good channel codes are ones which allow the successful detection and correction of transmission errors at coding rates R that are as close as possible to the theoretical limit that is imposed by the channel capacity [1]. In the past couple of decades, several near-capacity channel codes have emerged, including the turbo codes that are adopted in 3G and 4G mobile broadband standards, the Low Density Parity Check (LDPC) codes [2] that are adopted in WiFi standards [3], as well as the more recent polar codes [4]. Turbo and LDPC codes employ an iterative decoding process, in which each successive decoding attempt informs the next, until the process converges. By contrast, polar codes use a successive cancellation decoding process, in which the decoding of each successive bit informs the decoding of the next. Since the channel decoder must overcome the uncertainty introduced by noise, interference and poor signal strength, it typically has a much greater complexity than the channel encoder. Owing to this, it is the channel decoder that is typically the main concern when designing a channel code.

II. What is driving the discussion for 5G?

While turbo codes have enabled high-performance communication in 3G and 4G cellular standards, a myth has emerged which questions their applicability to 5G. More specifically, the myth is that turbo codes have an inherently serial structure [5], which requires the steps of the turbo decoding process to be completed one at a time and in the right order. This is in contrast to LDPC codes, which are recognized as having an inherently parallel structure [5], allowing all steps of the decoding process to be completed simultaneously and within less time. If this myth were true, then it would be impossible to construct turbo decoders that achieve high decoded throughputs, measured in billions of bits per second (Gbps). Since all received bits must pass through the channel decoder, turbo decoders would therefore impose a severe bottleneck on 5G, which is targeting transmission throughputs of up to 20 Gbps [6]. However, at AccelerComm, we have dispelled this myth by demonstrating a turbo decoder that achieves a decoded throughput of 21.9 Gbps [7]. This has been achieved by redesigning the turbo decoder from the ground up, at both the algorithm and hardware level. In contrast to the conventional approach that has been adopted for more than two decades, AccelerComm's turbo decoder enables the same fully-parallel degree of processing that can be achieved for LDPC decoders. We have therefore demonstrated that turbo codes can be applied in 5G, without imposing a bottleneck upon the achievable transmission throughput.

Furthermore, another myth has emerged that turbo codes have greater complexity than LDPC or polar decoders. While it is true that turbo decoders require more computations than LDPC or polar decoders at most coding rates R, this white paper shows that it is the overall implementation complexity that really matters and that this depends on more than just a decoder's computation complexity. More specifically, a decoder's processing hardware resource requirement and energy consumption depends also on the complexity of the hardware interconnections that are required to meet the 5G requirement of flexibly supporting a wide variety of block lengths K and coding rates R. Owing to their low interconnect complexity and their inherent flexibility, we show that turbo codes facilitate high-throughput flexible channel coding at lower implementation complexities than both LDPC and polar codes.

Owing to the persistence of the above-described myths, the 3GPP standardization group is currently considering the replacement of turbo codes with LDPC or polar codes in the 5G 'new radio'. The rest of this white paper examines the 5G requirements for channel coding and discusses how they are met by each of turbo, LDPC and polar codes. We conclude that turbo codes are able to meet the 5G requirements to a greater degree than LDPC and polar codes, while offering the industry some very significant cost savings by offering backwards compatibility with 3G and 4G. It is for these reasons that AccelerComm is promoting the turbo code for 5G.

III. What are the requirements for 5G?

Like 3G and 4G, the aim of the 5G 'new radio' is to continue the trend of offering exponentially greater user experience and more diverse applications for cellular communications. However, this will impose greater requirements upon the channel code [8], as summarized in the following discussions.

A. Throughput

A peak transmission throughput of 20 Gbps is targeted for the 5G 'new radio', which is much higher than the 1 Gbps achieved by 4G. During video streaming, this significantly improved throughput will enable opportunistic forward buffering when the channel conditions are favorable, for example. This will substantially increase the efficiency and reliability of streamed video, which currently suffers from drop outs when the channel conditions become unfavorable.

Since all received data must pass through the channel decoder, it must offer a decoded throughput of $T = 20$ Gbps in order to avoid imposing a bottleneck. Achieving this throughput will require very

parallel processing. If we (perhaps optimistically) assume that $I = 10$ decoding steps (namely iterations or successive cancellation steps) are required and that the processors can run at a clock frequency of $F = 500$ MHz, then at least $P = I \cdot T/F = 400$ parallel processors will be required. This parallelism may be implemented internally to give a fast channel decoder, which achieves a high decoded throughput by using an array of P processors that work together on the processing of each block. Alternatively, a high decoded throughput can be achieved using external parallelism, where multiple slow channel decoders are used to process multiple blocks at the same time, or where multiple blocks are 'unrolled' and pipelined through the same decoder at the same time [9], [10]. However, this external parallelism approach does not achieve the latency requirements of 5G, as discussed in the following Section III-B.

B. Latency

An end-to-end latency of 0.5 ms is targeted for the 5G 'new radio', which is much lower than the 10 ms achieved in 4G. This significantly improved latency will allow user inputs made on a handset to be delivered to the cloud, processed on a cloud computer and then returned to update the display of the handset, without the user noticing any delay. This will enable new applications in user-specific 3D video rendering, augmented reality, remote control and mobile gaming. Furthermore, since machines are more sensitive to delay than humans, these ultra-low latencies will enable new applications of machine-to-machine communication, such as swarm robotics, factory automation, as well as vehicular efficiency and safety.

However, an end-to-end latency of 0.5 ms implies a physical layer latency of 50 μs [11]. Furthermore, the channel decoder must share this latency budget with many other physical layer components, some of which must necessarily impose a relatively high latency, such as synchronization. Owing to this, the channel decoder should target the lowest possible processing latency. This is achieved by fast channel decoders, which comprise different processors that work together on the processing of each block. This is analogous to using a fast airplane to make several trips per day, delivering a high number of passengers per day (a high throughput), as well as a short trip duration (a low latency). Using this approach, a decoded throughput of $T = 20$ Gbps for data blocks comprising $K = 10000$ bits implies a processing latency of $L = K/T = 0.5$ μs, which is sufficiently small to enable an end-to-end latency of 0.5 ms. Note that while a high throughput can be achieved by using multiple slow channel decoders to process multiple blocks at the same time, each individual channel decoder is still slow and so has a high latency. This is analogous to using a fleet of slow airplanes that each make one trip per day, achieving a high number of passengers per day (a high throughput), but giving a long trip duration (a high latency).

C. Error correction capability

The target for the 5G 'new radio' is for only 1 block in every 100,000 to suffer from communication errors that cannot be corrected by the channel decoder, when the communication link between the handset and basestation is of reasonable quality. This represents a significant improvement to the 1 block in every 10,000 that suffer from communication errors in 4G. This improved error correction requirement complements the improved latency requirement of Section III-B, since it makes it ten times less likely that the receiver will need to use Hybrid Automatic Repeat reQuest (HARQ) to request a retransmission of erroneously decoded data, which imposes a significant additional delay. Despite this however, HARQ will remain a vital component of 5G, in order to enable error-free communication.

D. Flexibility

Flexibility is a key requirement for the 5G 'new radio', owing to the much wider range of use cases that it is targeting, relative to previous generations. Alongside enhanced broadband data and telephony

services, these applications include the Internet of Things (IoT), vehicular communications and cloud computing. Owing to this, the channel code must support a wide variety of data block lengths K, as well as a wide variety of coding rates $R = K/N$. For example, short block lengths may be expected to be typical in IoT applications, while long block lengths are typical in broadband data and cloud computing applications. Low coding rates will be required in rural areas where basestations are deployed sparsely, while high coding rates may be used for ultra-dense urban deployments. If the channel code does not support a wide variety of data block lengths K, then each data block may need to be padded with a high number of wasteful dummy data bits, such that its length becomes one of those supported by the channel code. Likewise, if the channel code does not support a wide variety of coding rates R, then it may be necessary to select one that is much lower than is really required by the current level of noise, interference and signal strength. This implies the transmission of a high number of wasteful encoded bits. Either way, both wasteful data bits and wasteful encoded bits are manifested as wasted spectrum. More specifically, the waste results in each transmission having a higher bandwidth, duration or energy than is really needed, preventing the spectrum from being used by other users at the same frequency, time or location, without suffering enhanced interference. This will therefore degrade the throughput, latency and error correction capability that can be offered by 5G. For this reason, the 5G flexibility requirement is key to maximally fulfilling the other challenging 5G requirements discussed in Sections III-A – III-C.

E. Implementation complexity

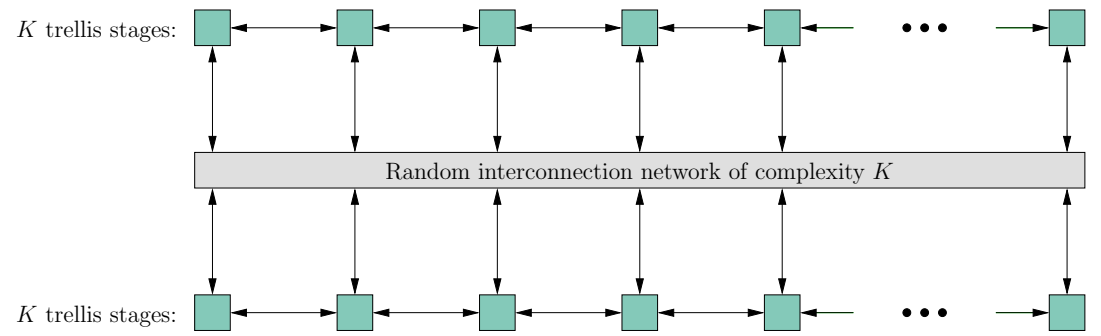
The implementation complexity of a channel decoder dictates its hardware resource requirement and energy consumption. At the time of writing, specific hardware and energy requirements have not been agreed for the 5G 'new radio'. However, it may be expected that the hardware and energies efficiencies of the channel decoder will be required to be at least as good as those of 4G. Here, the hardware efficiency quantifies a channel decoder's throughput as a ratio to its Application Specific Integrated Circuit (ASIC) area, which is measured in $\mathrm{Mbps/mm^2}$. Meanwhile the energy efficiency quantifies the number of bits that may be decoded per nJ of energy consumed by the channel decoder.

IV. How do turbo codes compare with LDPC and polar codes?

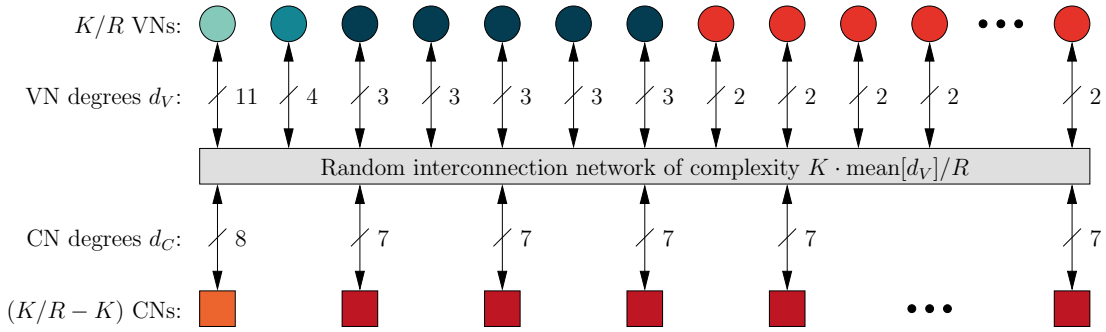
In this section, we compare turbo, LDPC and polar codes. Our discussions focus on the decoders, since these have much greater complexity than the corresponding encoders, therefore deciding the degree to which each of these three codes can meet the 5G requirements of Section III. Our discussions make references to Figure 1, which depicts the structure of the turbo, LDPC and polar decoders. Our comparisons are summarized in Table I and discussed in the following subsections.

A. Maturity

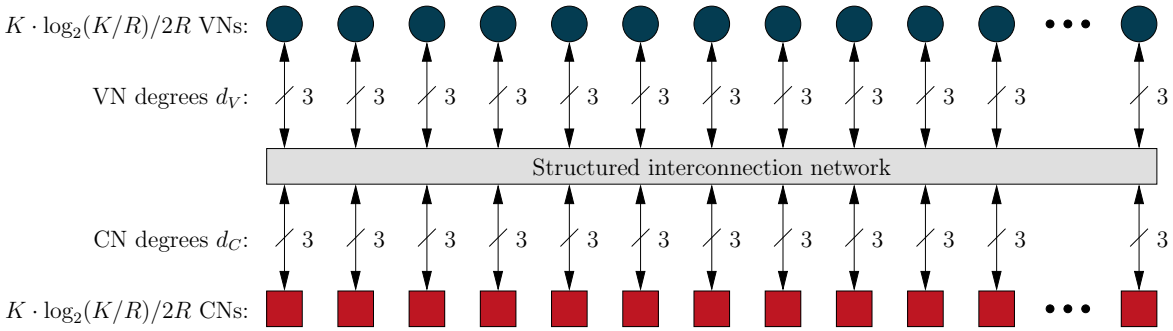
While turbo and LDPC codes entered the consciousness of the communications community in 1993 [12] and 1996 [2] respectively, polar codes were not proposed until much more recently in 2009 [4]. Owing to this, turbo and LDPC codes have reached a much greater level of maturity than polar codes, as shown in Table I. In particular, turbo and LDPC codes can be found in many consumer devices, owing to their inclusion in 3G/4G and WiFi standards, respectively. By contrast, polar codes have not yet been adopted in any standards or consumer devices and so their maturity is limited to proof of concept demonstrators and academic publications.



(a) Turbo decoder structure



(b) LDPC decoder structure



(c) Polar decoder structure

Fig. 1. Structures of turbo, LDPC and polar decoders.

TABLE I
COMPARISON BETWEEN TURBO, LDPC AND POLAR DECODERS.

Characteristic	Turbo decoder	LDPC decoder	Polar decoder
Maturity	Proven in 3G and 4G	Proven in WiFi	Unproven
Throughput and latency	Proven by AccelerComm	Proven for fully-parallel	Proven for pipelining
Error correction capability	Similar	Similar	Similar for list decoding
Flexibility	Flexible	Inflexible	Inflexible
Computation complexity	Higher for most coding rates	Lower for most coding rates	Lower for most coding rates
Interconnect complexity	Lower	Higher	Lower
High-performance flexible implementation complexity	Lower	Higher	Unproven
Backwards compatibility	Yes	No	No

B. Throughput and latency

As shown in Table I, decoded throughputs exceeding 20 Gbps together with correspondingly fast latencies have been demonstrated for each of turbo, LDPC and polar decoders, albeit with limited flexibility and/or degraded error correction capability. As described in Section II, AccelerComm has previously demonstrated a fully-parallel turbo decoder that achieves a decoded throughput of 21.9 Gbps and a processing latency of 0.24 μs, for a turbo code having a block length of $K = 6144$ and a coding rate of $R = 1/3$ [7]. While this previous design has only a limited flexibility, AccelerComm now has a new design that is fully flexible and fully compatible with the 4G Long Term Evolution (LTE) turbo code, as discussed in Section IV-D.

Likewise, LDPC decoders are required to adopt fully-parallel processing in order to achieve decoded throughputs in excess of 20 Gbps. In particular, a decoded throughput of $T = 78$ Gbps and a latency of 0.06 μs have been demonstrated in [13] for an LDPC code having a block length of $K = 1723$ and a coding rate of $R = 0.84$. However, the LDPC decoder of [13] supports only this single combination of block length K and coding rate R, which speaks to the poor flexibility of LDPC decoders in general and of fully-parallel LDPC decoders in particular, as discussed in Section IV-D. Note that like many publications of LDPC decoders, [13] does not directly quantify the decoded throughput T, instead giving only the higher value of the encoded throughput T/R.

The only polar decoders that achieve a decoded throughput in excess of 20 Gbps have managed this by unrolling hundreds of data blocks and pipelining their successive cancellation decoding through the same hardware [10]. Using this approach, an extremely high throughput of 208 Gbps has been demonstrated for a coding rate of $R = 1/2$, although the latency experienced by each $K = 1024$-bit data block is 3.21 μs, for the reasons discussed in Section III-B. However, polar decoders are particularly inflexible and this pipelining approach limits the flexibility even further, as discussed in Section IV-D. Furthermore, the successive cancellation technique for polar decoding offers only a degraded error correction capability, compared to the more complex list decoding technique, as will be discussed in Section IV-C.

C. Error correction capability

As described in Section I, turbo, LDPC and polar codes are all examples of near-capacity codes. Owing to this, they are all capable of offering the same error correction capability [14], albeit with different complexities, as will be discussed in Sections IV-E and IV-F. Indeed, a plethora of different approaches to the decoding of each type of code have been proposed, with some offering significantly reduced complexity, at the cost of only slightly degraded error correction capability. As described in Section I, turbo and LDPC decoders employ an iterative decoding process. While this approach can be employed for polar decoding, the unique appeal of polar decoding is that they can achieve near-capacity operation,

without the requirement for iterative decoding. Instead, polar codes can use a successive cancellation decoding process [4], which significantly reduces the decoding complexity, at the cost of degrading their error correction capability. This degradation can be recovered however, by employing the list decoding technique [15]. Rather than considering only the most likely decoding decisions, the list decoding technique instead dynamically selects and considers the L number of most likely decoding decisions. However, list decoding increases the complexity of the polar decoder by L times, as will be discussed in Section IV-E. Nonetheless, list decoding allows polar decoders to offer a similar error correction capability to turbo and LDPC decoders, as shown in Table I

As described in Section III-C, a requirement of 5G is to offer a ten times improvement upon the error correction capability of the 4G LTE turbo code. As described in [16], this may be achieved by redesigning several details in the operation of the turbo decoder, which are referred to as termination, puncturing and interleaving. Note that while redesigning the interleaving may invalidate the conventional solution to the so-called contention problem during turbo decoding, AccelerComm's flexible turbo decoder does not suffer from contention for any interleaver design. Similarly, LDPC and polar codes can be specifically designed to meet the 5G requirement for error correction capability.

D. Flexibility

The flexibility of turbo, LDPC and polar decoders to support various combinations of block length K and coding rate $R = K/N$ is dictated by their structures, which are illustrated in Figure 1. Turbo decoders have inherent flexibility because their structure comprises two interconnected rows of K identical so-called trellis stages. This uniformity among the trellis stages reflects the equal error protection that is applied to each of the data bits during turbo encoding. Owing to this, turbo codes are well suited to HARQ, as well as puncturing and repetition techniques [17], which allow them to flexibly support any coding rates R. Furthermore, the 4G LTE turbo code exploits the regularity of the structure depicted in Figure 1a to flexibly support 188 different block lengths K in the range 40 to 6144 bits. This regularity allows a number P of identical parallel processors to work together to complete the decoding of each block. It is in this way that the AccelerComm turbo decoder can apply hundreds or thousands of parallel processors, while flexibly supporting any sets of block lengths K, having any interleaver designs, as described in Section IV-C.

In comparison to turbo decoders, the structure of LDPC decoders is much less regular, as shown in Figure 1b. More specifically, LDPC decoders resemble a random graph of N so-called Variable Nodes (VNs), which are interconnected with $(N-K)$ so-called Check Nodes (CNs). The computations associated with VNs and CNs are different to each other, as are the computations associated with nodes having different numbers of connections to other nodes, where these numbers are referred to as the degrees of the nodes. This irregularity reflects the unequal error protection that is applied to the data bits during LDPC encoding, which makes LDPC codes less suited to HARQ, puncturing and repetition techniques. Indeed, HARQ is typically achieved by using very large graphs, where each retransmission conveys a different subset of the encoded bits, although this results in a high computational complexity as discussed in Section IV-E. Likewise, flexibility to use different coding rates R and different block lengths K is typically achieved by using a different graph for each supported combination, where the graphs have different node degrees in different distributions. This makes it difficult to efficiently reuse parallel processors to support a wide variety of graphs, particularly when fully-parallel processing is required. Owing to this, the WiFi LDPC code supports only 12 combinations of coding rate R and block length K, while the most flexible standardized LDPC code is that of WiMAX, which supports 76 combinations [18]. Both the WiFi and WiMAX LDPC codes use a so-called quasi-cyclic graph construction, which allows the reuse of parallel processors. However, the degree of parallelism P that can be readily achieved by quasi-cyclic LDPC

decoders is limited by their so-called expansion factor Z. In the WiFi LDPC code for example, the expansion factor adopts values with a greatest common divisor of $Z = 27$, which is much lower than the parallelism of $P = 400$ that is required to achieve a decoded throughput of 20 Gbps, as described in Section III-A. Owing to these issues, LDPC codes are relatively inflexible, particularly when a high decoded throughput is required, as shown in Table I.

As shown in Figure 1c, polar decoders have a very regular structure, which comprises a structured graph of VNs and CNs, all having the same degree. However, the design of a polar code must be optimized not just for each supported combination of coding rate R and block length K, but also for each channel condition. This is because in contrast to the iterative turbo and LDPC decoding processes, polar decoders cannot generally use bits that are less affected by noise, interference or poor signal strength to help recover the bits that are more severely affected. Since memory is required to store the parameters of the designs optimized for each combination of coding rate, block length and channel condition, only a limited number of combinations can be supported in practice. Note that as described in Section IV-B, the 5G throughput requirements have so-far only been met by the polar decoders of [10], which pipeline the successive cancellation decoding of hundreds of data blocks through the same hardware. However, the hardware of [10] is particularly inflexible, since it requires every pipelined block to have the same coding rate R, block length K and channel conditions. In particular, it is not compatible with the dynamic decisions made by the list decoding technique, which is required in order to meet the error correction capability requirements of 5G. Owing to these issues, polar codes are relatively inflexible, particularly when a high decoded throughput and a strong error correction capability is required, as shown in Table I.

E. Computation complexity

The computations performed by most practical implementations of turbo, LDPC and polar decoders comprise mainly maximum, minimum and addition operations (MaxMinAdd) [19], [20]. Since these operations all have similar complexity, the computation complexity of turbo, LDPC and polar decoders can be easily compared by quantifying the number of MaxMinAdd operations that are required to decode each bit.

In the 4G LTE turbo decoder, a total of 155 MaxMinAdd operations are required per data bit per iteration [7]. As few as 6 iterations are required when a low degree of parallelism is employed and around 28 iterations are required for fully-parallel turbo decoding. This results in a computation complexity of up to 4340 MaxMinAdd operations per data bit, irrespective of the block length K or the coding rate R.

Meanwhile, the computation complexity of an LDPC decoder is given by $(6 \cdot \text{mean}[d_v] - 9)/R + 6$ MaxMinAdd operations per data bit per iteration [21], where the mean of the VN degrees is given by $\text{mean}[d_v] \approx 3.5$ in the case of all 12 WiFi LDPC codes, for example. In common with turbo decoders, the number of decoding iterations required depends on the parallelism of the LDPC decoder, where similar numbers of iterations are required in order to achieve the same error correction capability of turbo codes. Owing to this, the computation complexity of an LDPC decoder can be as high as 840 MaxMinAdd operations per data bit when performing $I = 28$ iterations in the case of the lowest coding rate $R = 1/2$ supported by the WiFi LDPC decoder. Likewise, its highest coding rate of $R = 5/6$ gives computation complexities as high as 560 MaxMinAdd operations per data bit when performing $I = 28$ iterations. Note that the computation complexity of LDPC decoders varies significantly with the coding rate R, although it is lower than that of turbo decoders for all but the lowest coding rates, as shown in Table I. Note however that low coding rates may be necessary in order to implement HARQ for LDPC codes, as described in Section IV-D. In this case, a high computation complexity is imposed when decoding each retransmission, which conveys a different subset of the encoded bits.

In a polar decoder employing successive cancellation, the computation complexity is given by $\log_2(K/R)/R$ MaxMinAdd operations per data bit [4]. In the case of $R = 1/2$ and $K = 4096$, we have 26 MaxMinAdd operations per data bit, which is significantly lower than in turbo and LDPC decoders. However, the use of the successive cancellation technique results in an excessive degradation in error correction capability, as described in Section IV-C. This degradation can be recovered by using the list decoding technique, although this increases the computation complexity by as many as $L = 32$ times, making the complexity of polar decoders comparable to those of LDPC decoders, as shown in Table I.

F. Interconnect complexity

As shown in Figure 1, turbo, LDPC and polar decoders all rely on interconnection networks, which can add significant complexity to the implementation of flexible decoders. In particular, the complexity of these interconnection networks depends on the flexibility offered for the data block length K, as well as the coding rate R in the case of LDPC decoders. These interconnection networks can require information generated on one side of an ASIC implementation to be routed all the way to the opposite side of the chip. Furthermore, the design of the interconnection network has significant impacts on the memory management that is required in a hardware implementation, in order to avoid the contention problem that was mentioned in Section IV-C. While the implementation of the computations performed by turbo, LDPC and polar decoders is relatively straightforward, it is the implementation of the interconnection networks that imposes the real challenge. The complexity of an interconnection network is determined by the amount of information that it must route, as well as by its randomness.

The interconnection network of a turbo code is comprised by its interleaver, which has a relatively random design. However, the parametrization of the 4G LTE turbo code interconnection network for all 188 block lengths K is described by a single equation and a single page of parameters in the standard document [17]. As shown in Figure 1a, the complexity of the turbo code interconnection network is simply given by the block length K, since this determines the amount of information that is exchanged through the interleaver. Note that while some information is also exchanged between neighboring trellis stages, the complexity associated with this is very small since the processors of neighboring trellis stages can be readily positioned adjacently on a turbo decoding hardware implementation.

By contrast, the interconnection network of an LDPC decoder is comprised by its graph, which has a particularly large and random design, leading to a high interconnect complexity, as shown in Table I. In contrast to the 4G LTE turbo code, the parametrization of the WiFi LDPC code interconnection network for all 12 supported combinations of block length K and coding rate R occupies three pages in the standard document [3]. As shown in Figure 1a, the complexity of the LDPC interconnection network is given by $K \cdot \text{mean}[d_v]/R$, where $\text{mean}[d_v] \approx 3.5$ in the case of the WiFi LDPC code. Note that in the case of a coding rate of $R = 1/2$, the LDPC interconnect complexity is seven times greater than that of a corresponding turbo decoder and is even greater for lower coding rates. This has a very significant effect on the overall implementation complexity of LDPC decoders, as discussed in Section IV-G.

As shown in Figure 1c, the interconnection network of a polar decoder is large but it adopts a regular structure, which is relatively simple to implement. Owing to this, the interconnect complexity of a polar decoder is relatively low, as shown in Table I.

G. High-performance flexible implementation complexity

While the computation and interconnect complexities of Sections IV-E and IV-F are useful for understanding the differences between turbo, LDPC and polar decoders, it is the implementation complexity that is of real importance. This is because the implementation complexity is what determines the extent to which

a decoder meets the 5G requirements for hardware and energy efficiency, as discussed in Section III-E. In particular, we should be interested in the implementation complexity of decoders that come closest to meeting the high-performance 5G requirements of Sections III-A – III-C for throughput, latency and error correction, as well as the versatile 5G flexibility requirements of Section III-D.

At the time of writing, a comprehensive comparison of flexible ASIC implementations of turbo, LDPC and polar decoders is not available. Indeed, no ASIC polar decoder implementations have been proposed that support the flexible list decoding of blocks having various lengths K and coding rates R. This highlights the immaturity of polar codes, as shown in Table I. However, it is useful to compare the highest throughput ASIC implementation of the LTE turbo decoder with the highest throughput ASIC implementation of the most flexible standardized LDPC decoder, namely that of WiMAX. More specifically, the turbo decoder of [22] has a 65 nm chip area of 7.7 mm^2 and achieves a decoded throughput of 2.15 Gbps, while supporting all 188 LTE block lengths K and any coding rate R. By comparison, the LDPC decoder of [23] supports all combinations of the 19 WiMAX block lengths K and the 4 WiMAX coding rates K. It has a 65 nm chip area of 3.36 mm^2 and achieves an encoded throughput of 1.06 Gbps, which corresponds to a decoded throughput of 0.53 Gbps when the coding rate is $R = 1/2$. Despite having significantly less flexibility and a lower throughput, the LDPC decoder of [23] achieves a poorer hardware efficiency of 158 Mbps/mm^2, as compared with the 279 Mbps/mm^2 of the turbo decoder of [22]. This is contrary to the expectations set by the computation complexity comparison of Section IV-E, but may be explained by the LDPC decoder's significantly higher interconnect complexity, as characterized in Section IV-F. Unfortunately, [22] does not quantify the energy efficiency of this turbo decoder ASIC and so it cannot be compared with that of the LDPC decoder ASIC of [23].

Furthermore, Field Programmable Gate Array (FPGA) implementations of flexible turbo and LDPC decoders were comprehensively compared in [24], [25], following a close-to exhaustive search of academic papers and commercial data sheets. Note that while the programmable general-purpose nature of FPGAs means that they are unable to achieve the decoded throughputs offered by specialized ASICs, this also means that they enable the rapid prototyping of new ideas, allowing a comprehensive and timely comparison. Note that a much higher number of FPGA implementations of LDPC decoders have been produced than turbo decoders. At the time of writing, no FPGA polar decoder implementations have been proposed that support flexible list decoding, again highlighting the immaturity of polar codes.

Figure 2 summarizes the flexible turbo and LDPC decoder comparisons of [24], [25], in terms of decoding throughput, hardware usage and error correction capability. Of particular note are the three cyan triangles across the top of the scatter, which represent three different implementations of the 4G LTE turbo decoder, namely the Altera reference design [26], the Xilinx reference design [27] and the academic design of [28]. These implementations offer superior decoded throughput, hardware usage and/or error correction capability to nearly all of the flexible LDPC decoders. Again, this is contrary to the expectations set by the computation complexity comparison of Section IV-E, but may be explained by the LDPC decoders' significantly higher interconnect complexity, as characterized in Section IV-F. Furthermore, most of the LDPC decoders characterized in Figure 2 implement the WiFi LDPC code, which offers significantly less flexibility than the 4G LTE turbo code, as discussed in Section IV-D.

Note that the complexity and hence decoded throughput of a turbo decoder does not change with the coding rate R, owing to its use of the puncturing technique. By contrast, Section IV-E showed that reducing the coding rate of an LDPC decoder increases its complexity, which may be expected to reduce its decoded throughput. This gives some context to the few LDPC decoders that sit above the line formed by [26]–[28]. These LDPC decoders have relatively poor error correction capability owing to their employment of high coding rates. More specifically, the error correction capability of these decoders could be improved by reducing their coding rate towards that of the 4G LTE turbo code, but this would increase their complexity

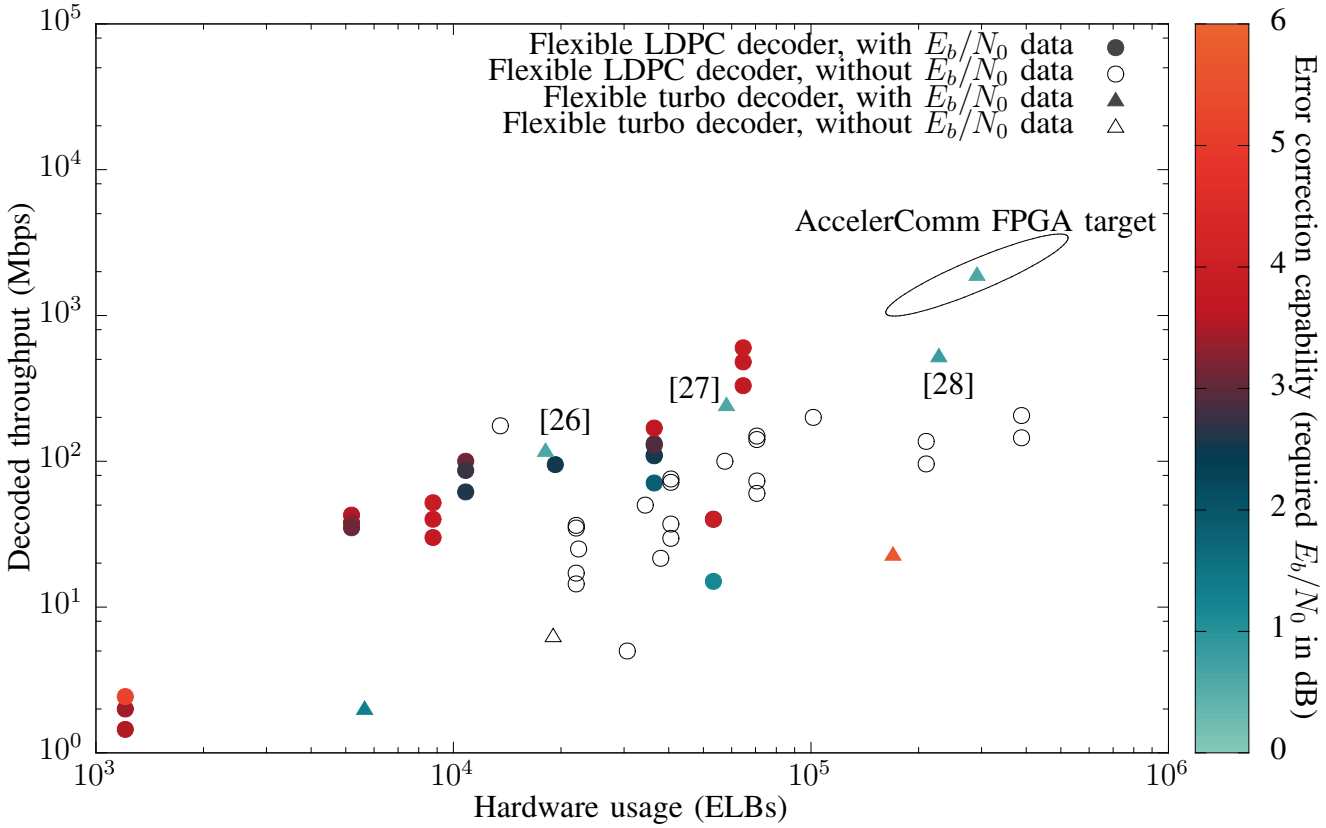


Fig. 2. A comparison between the FPGA implementations of various flexible turbo and LDPC decoders, in terms of decoded throughput, hardware resource requirements and error correction capability. References for each data point can be found in the annotations of the corresponding figures in [24], [25]. The hardware usage is quantified using Equivalent Logic Blocks (ELBs), as defined in [24]. The error correction capability is quantified by the SNR per bit E_b/N_0 in dB required to achieve a BER of 10^{-4}.

and reduce their throughput to below the line formed by [26]–[28]. With this in mind, we may conclude the that best flexible turbo decoders available today have a similar implementation complexity as the best flexible LDPC decoders, albeit the latter have less flexibility. If the flexibility of these LDPC decoders was increased to meet the requirements of 5G, then it may be expected that their implementation complexity would increase significantly above that of the best flexible turbo decoders, as shown in Table I. This would result in an inferior hardware and energy efficiency for LDPC decoders, relative to turbo decoders.

H. Backwards compatibility

In order to ensure that 5G can benefit from all of the most recent advances that have been made in wireless communications research, the 5G 'new radio' is being developed on a blank slate, with no requirement for backwards compatibility with 4G or earlier. However in the case of all other things being equal, an option that is backwards compatible may be considered to offer a significant advantage. This is because, just as how 4G handsets must also support 2G and 3G in order to achieve acceptable coverage, 5G handsets will also need to support 3G and 4G. If 5G adopts the turbo code, then handsets can be designed to use the same hardware for all of 3G, 4G and 5G turbo decoding. By contrast, if another channel code is selected for 5G, then handsets will be required to have two separate pieces of hardware, namely one for 3G and 4G turbo decoding, as well as another for 5G channel decoding. This would represent a significant overhead, since the channel decoder occupies up to 72% of a typical baseband

processor hardware resource requirement and energy consumption [29]. Even if there is only a small financial cost associated with including this 3G/4G turbo decoder in each 5G handset, this potentially aggregates to an enormous sum across the billions of devices that may be expected in the lifetime of 5G. Furthermore, since turbo codes have been adopted in 3G and 4G, they offer a short time to market for 5G, with only a modest development cost, in contrast to the long time to market and large development costs that would be associated with LDPC or polar codes. Owing to their backwards compatibility, the adoption of turbo codes in 5G would save the cellular communications industry and ultimately its consumers from some very significant costs, as shown in Table I.

V. So what's next?

In this white paper, we have discussed the channel code requirements of 5G, which include high throughput, low latency, strong error correction capability and low implementation complexity. We have also highlighted a 5G requirement for the channel code to flexibly support a wide variety of block lengths K and coding rates R, in order to ensure efficient use of spectrum and to address the many use cases of 5G. We have provided a comprehensive discussion of the extent to which the 5G requirements can be met by each of turbo, LDPC and polar codes. In particular, we have dispelled two myths that have been the genesis of the debate over the selection of channel codes for 5G, namely that turbo codes cannot support high throughputs and that they have higher complexities than LDPC and polar codes. Indeed, AccelerComm has already demonstrated that turbo codes can achieve decoded throughputs exceeding the 5G target of 20 Gbps. Furthermore, we have shown that overall implementation complexity of a channel code depends not only on its computation complexity, but also its interconnect complexity and its inherent flexibility. We have shown that turbo codes hold the greatest promise for offering high performance throughputs, latencies and error correction capabilities, as well as high degrees of flexibility at the lowest implementation complexity. This may be attributed to the regular and flexible structure of turbo codes, which we contrasted with the relatively irregular and/or inflexible structures of LDPC and polar codes. Furthermore, we highlighted that turbo codes can offer the advantage of backwards compatibility to 3G and 4G, which may save the cellular communications industry from some very significant costs.

However, there is still a significant amount of work that must be completed before the 5G crown can be awarded to one of the channel code contenders. There is a need for further work on flexible, high-performance implementations of turbo, LDPC and list polar decoders. A comprehensive comparison of these implementations is also required, to extend those of [24], [25]. Our subsequent white papers will contribute to these ongoing efforts. In particular, we will detail the FPGA and ASIC implementations of our flexible, high performance turbo decoder. As shown in Figure 2, our FPGA implementation will target significantly higher decoded throughputs than have been achieved previously, while flexibly supporting all block lengths K and coding rates R of the 4G LTE turbo code. Our ASIC implementation will maintain this flexibility, while also targeting all other 5G requirements, as detailed in Section III. Furthermore, we will comprehensively compare our implementations with the state-of-the-art implementations found in academic publications and commercial data sheets.

References

[1] C. E. Shannon, "The mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 457–458, Aug. 1996.

[3] *IEEE 802.11-2012 Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*, 2012.

[4] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[5] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge University Press, 2009.
[6] *TR 38.913 Study on scenarios and requirements for next generation access technologies*, 3GPP Std., Rev. 0.3.0, Mar. 2016.
[7] A. Li, L. Xiang, T. Chen, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "VLSI Implementation of Fully-Parallel LTE Turbo Decoders," *IEEE Access*, vol. 4, pp. 323–346, Jan. 2016.
[8] Samsung, "R1-162181 Channel coding for 5G new radio interface," in *3GPP TSG RAN WG1 #84bis*, Apr. 2016.
[9] N. Wehn, S. Scholl, P. Schläfer, T. Lehnigk-Emden, and M. Alles, *Challenges and Limitations for Very High Throughput Decoder Architectures for Soft-Decoding*. Cham: Springer International Publishing, 2015, pp. 7–31.
[10] P. Giard, G. Sarkis, C. Thibeault, and W. J. Gross, "Unrolled polar decoders, part I: hardware architectures," *Submitted to IEEE J. Select. Areas Commun.* [Online]. Available: http://arxiv.org/abs/1505.01459
[11] G. P. Fettweis, "The Tactile Internet – applications and challenges," *IEEE Veh. Technol. Mag.*, pp. 64–70, Mar. 2014.
[12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," in *Proc. IEEE Int. Conf. on Communications*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.
[13] T. Mohsenin, D. N. Truong, and B. M. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.
[14] Nokia, "R1-162897 Performance and complexity of turbo, LDPC and polar codes," in *3GPP TSG RAN WG1 #84bis*, Apr. 2016.
[15] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
[16] Orange, "R1-164635 Improved LTE turbo codes for NR," in *3GPP TSG RAN WG1 #85*, May 2016.
[17] *ETSI TS 136 212 LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding*, V13.1.0 ed., 2016.
[18] *IEEE 802.16-2012 Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Broadband Wireless Access Systems*, 2012.
[19] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. on Communications*, vol. 2, Seattle, WA, USA, June 1995, pp. 1009–1013.
[20] M. Fossorier, "Reduced complexity decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.
[21] G. D. Forney, "The forwards-backwards algorithm," in *Proc. Allerton Conf. on Communications, Control, and Computing*, Monticello, IL, USA, Oct. 1996, pp. 432–446.
[22] T. Ilnseher, F. Kienle, C. Weis, and N. Wehn, "A 2.15GBit/s turbo code decoder for LTE Advanced base station applications," in *Proc. Int. Symp. on Turbo Codes and Iterative Information Processing*, Gothenburg, Sweden, Aug. 2012, pp. 21–25.
[23] X. Peng, Z. Chen, X. Zhao, D. Zhou, and S. Goto, "A 115mW 1Gbps QC-LDPC decoder ASIC for WiMAX in 65nm CMOS," in *Proc. IEEE Asian Solid State Circuits Conf.*, Jeju, Korea, Nov 2011, pp. 317–320.
[24] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A survey of FPGA-based LDPC decoders," *IEEE Commun. Surveys Tutorials*, vol. 18, no. 2, pp. 1098–1122, Second quarter 2016.
[25] A. Li, P. Hailes, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "1.5 Gbit/s FPGA implementation of a fully-parallel turbo decoder designed for mission-critical machine-type communication applications," *IEEE Access (Submitted)*. [Online]. Available: http://eprints.soton.ac.uk/399185/
[26] Altera, "Turbo IP Core - User Guide," 2015.
[27] Xilinx, "LogiCORE IP 3GPP Mixed Mode Turbo Decoder v2.0," 2014.
[28] L. F. Gonzalez-Perez, L. C. Yllescas-Calderon, and R. Parra-Michel, "Parallel and configurable turbo decoder implementation for 3GPP-LTE," in *Proc. Int. Conf. Reconfigurable Computing and FPGAs*, Cancun, Mexico, Dec. 2013, pp. 1–6.
[29] H. K. Boyapati, R. V. Rajakumar, and S. Chakrabarti, "Quantifying the improvement in energy savings for LTE eNodeB baseband subsystem with technology scaling and multi-core architectures," in *Proc. Nat. Conf. on Commun.*, Kharagpur, India, Feb. 2012, pp. 1–5.

Robert G. Maunder is an industry authority on channel coding, having gained his PhD in Wireless Communications in 2007 from Southampton University. As an Associate Professor, he published over 100 IEEE papers and resources on the holistic design of signal processing algorithms and hardware implementations for error correction decoders, including turbo, LDPC and polar decoders. This work led him to dispel the longstanding myth that the level of parallelism in turbo decoders is inherently limited. Rob has found a way to unlock unlimited levels of parallelism and with it, dramatically improve turbo decoder throughput and latency. This is what is being leveraged by his founding of AccelerComm Limited.