

METHODS AND APPARATUS FOR COMPACTLY DESCRIBING LIFTED LOW-DENSITY PARITY-CHECK (LDPC) CODES

Cross-Reference to Related Applications & Priority Claim

[0001] This application claims benefit of and priority to U.S. Provisional Patent Application Serial No. 62/349,784, filed June 14, 2016 (163764P1), and to U.S. Provisional Patent Application Serial No. 62/374,514 (164403P1), filed August 12, 2016, and U.S. Patent Application No. 15/622,019, filed June 13, 2017 (163764), filed June 13, 2017, all three of which are herein incorporated by reference in their entirety for all applicable purposes.

TECHNICAL FIELD

[0002] Certain aspects of the technology discussed below generally relate to wireless communications, including detecting and/or correcting errors in binary data and, more particularly, to methods and apparatus for compactly describing lifted low-density parity-check (LDPC) codes.

INTRODUCTION

[0003] Wireless communication systems are widely deployed to provide various types of communication content such as voice, video, data, message, broadcasts, and so on. These systems may employ multiple-access technologies capable of supporting communication with multiple users by sharing available system resources (e.g., bandwidth and transmit power). Examples of such multiple-access systems include code division multiple access (CDMA) systems, time division multiple access (TDMA) systems, time division synchronous CDMA (TD-SCDMA) systems, frequency division multiple access (FDMA) systems, single-carrier FDMA (SC-FDMA) systems, orthogonal FDMA (OFDMA), 3rd Generation Partnership Project (3GPP) long term evolution (LTE) systems, and LTE Advanced (LTE-A) systems.

[0004] Multiple access technologies have been adopted in various telecommunication standards to provide a common protocol that enables different wireless devices to communicate on a municipal, national, regional, and even global level. An example of an emerging telecommunication standard is new radio (NR), for example, 5G radio access. NR is a set of enhancements to the LTE mobile standard promulgated by 3GPP. It is

designed to better support mobile broadband Internet access by improving spectral efficiency, lowering costs, improving services, making use of new spectrum, and better integrating with other open standards using OFDMA with a cyclic prefix (CP) on the downlink (DL) and on the uplink (UL) as well as support beamforming, multiple-input multiple-output (MIMO) antenna technology, and carrier aggregation.

[0005] Generally, a wireless multiple-access communication system can simultaneously support communication for multiple wireless nodes. Each node communicates with one or more base stations (BSs) via transmissions on forward and reverse links. The forward link (or downlink) refers to a communication link from BSs to nodes, and a reverse link (or uplink) refers to a communication link from nodes to base stations. Communication links may be established via a single-input single-output, multiple-input single-output, or a MIMO system.

[0006] In some examples, a wireless multiple-access communication system may include a number of BSs, each simultaneously supporting communication for multiple communication devices, otherwise known as user equipment (UEs). In an LTE or LTE-A network, a set of one or more BSs may define an e NodeB (eNB). In other examples (e.g., in a next generation, NR, or 5G network), a wireless multiple access communication system may include a number of distributed units (DUs) (e.g., edge units (EUs), edge nodes (ENs), radio heads (RHs), smart radio heads (SRHs), transmission reception points (TRPs), etc.) in communication with a number of central units (CUs) (e.g., central nodes (CNs), access node controllers (ANCs), etc.), where a set of one or more DUs, in communication with a CU, may define an access node (e.g., a BS, a NR BS, a 5G BS, a NB, an eNB, NR NB, a 5G NB, an access point (AP),), a network node, a gNB, a TRP, etc.). A BS, AN, or DU may communicate with a UE or a set of UEs on downlink channels (e.g., for transmissions from a BS or to a UE) and uplink channels (e.g., for transmissions from a UE to a BS, AN, or DU).

[0007] Binary values (e.g., ones and zeros), are used to represent and communicate various types of information, such as video, audio, statistical information, etc. Unfortunately, during storage, transmission, and/or processing of binary data, errors may be unintentionally introduced; for example, a “1” may be changed to a “0” or vice versa.

[0008] Generally, in the case of data transmission, a receiver observes each received bit in the presence of noise or distortion and only an indication of the bit's value is obtained. Under these circumstances, the observed values are interpreted as a source of "soft" bits. A soft bit indicates a preferred estimate of the bit's value (e.g., a 1 or a 0) together with some indication of the reliability of that estimate. While the number of errors may be relatively low, even a small number of errors or level of distortion can result in the data being unusable or, in the case of transmission errors, may necessitate retransmission of the data. In order to provide a mechanism to check for errors and, in some cases, to correct errors, binary data can be coded to introduce carefully designed redundancy. Coding of a unit of data produces what is commonly referred to as a codeword. Because of its redundancy, a codeword will often include more bits than the input unit of data from which the codeword was produced.

[0009] Redundant bits are added by an encoder to the transmitted bitstream to create a codeword. When signals arising from transmitted codewords are received or processed, the redundant information included in the codeword as observed in the signal can be used to identify and/or correct errors in or remove distortion from the received signal to recover the original data unit. Such error checking and/or correcting can be implemented as part of a decoding process. In the absence of errors, or in the case of correctable errors or distortion, decoding can be used to recover from the source data being processed, the original data unit that was encoded. In the case of unrecoverable errors, the decoding process may produce some indication that the original data cannot be fully recovered. Such indications of decoding failure initiate retransmission of the data. As the use of fiber optic lines for data communication and the rate at which data can be read from and stored to data storage devices, (e.g., disk drives, tapes, etc.) increases, there is an increasing need for efficient use of data storage and transmission capacity and also for the ability to encode and decode data at high rates.

BRIEF SUMMARY

[0010] The following summarizes some aspects of the present disclosure to provide a basic understanding of the discussed technology. This summary is not an extensive overview of all contemplated features of the disclosure, and is intended neither to identify key or critical elements of all aspects of the disclosure nor to delineate the scope of any or all aspects of the disclosure. Its sole purpose is to present some concepts of one or more

aspects of the disclosure in summary form as a prelude to the more detailed description that is presented later. After considering this discussion, and particularly after reading the section entitled “Detailed Description” one will understand how the features of this disclosure provide advantages that include improved communications between access points and stations in a wireless network.

[0011] While encoding efficiency and high data rates are important, for an encoding and/or decoding system to be practical for use in a wide range of devices (e.g., consumer devices), it is also important that the encoders and/or decoders can be implemented at reasonable cost.

[0012] Communication systems often need to operate at several different rates. Low-density parity-check (LDPC) codes can be used for simple implementation to provide coding and/or decoding at different rates. For example, higher-rate LDPC codes can be generated by puncturing lower-rate LDPC codes.

[0013] As the demand for mobile broadband access continues to increase, there exists a need for further improvements in NR technology. Preferably, improvements can or should be applicable to other multi-access technologies and the telecommunication standards that employ these technologies. One area for improvements is the area of encoding/decoding for data transmissions. These improvements (e.g., improved LDPC codes) can be applicable to NR and other access technologies.

[0014] Certain aspects of the present disclosure generally relate to methods and apparatus for compactly describing multiple lifted low-density parity-check (LDPC) codes.

[0015] Certain aspects of the present disclosure provide a method for wireless communications that may be performed by a transmitting device. The method generally includes selecting a first lifting size value Z and a first set of lifting values for generating a first lifted LDPC code. A transmitting device can generate a first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base parity check matrix (PCM) having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code (an edge is a connection between a variable node and a check node). A transmitting device can determine a second set of lifting values for generating a second lifted PCM

corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values. A transmitting device can encode a set of information bits based on the first lifted LDPC code and/or the second lifted LDPC code to produce a code word and transmits the code word.

[0016] Certain aspects of the present disclosure provide an apparatus for wireless communications that may be performed by a transmitting device. The apparatus generally includes means for selecting a first lifting size value Z and a first set of lifting values for generating a first lifted LDPC code and means for generating the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base PCM having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code. The apparatus also includes means for determining a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values; means for encoding a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and means for transmitting the code word.

[0017] Certain aspects of the present disclosure provide an apparatus for wireless communications that may be performed by a transmitting device. The apparatus generally includes at least one processor coupled with a memory. The at least one processor is configured to select a first lifting size value Z and a first set of lifting values for generating a first lifted LDPC code and generate the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base PCM having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code. The at least one processor is also configured to determine a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values and encode a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word. The apparatus includes a transmitter configured to transmit the code word.

[0018] Certain aspects of the present disclosure provide a computer readable medium having computer executable code stored thereon for wireless communications that may be

performed by a transmitting device. The code generally includes code for selecting a first lifting size value Z and a first set of lifting values for generating a first lifted LDPC code and code for generating the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base PCM having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code. The code also includes code for determining a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values; code for encoding a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and code for transmitting the code word.

[0019] Other aspects, features, and embodiments of the present disclosure will become apparent to those of ordinary skill in the art, upon reviewing the following description of specific, exemplary aspects of the present disclosure in conjunction with the accompanying figures. While features of the present disclosure may be discussed relative to certain aspects and figures below, all aspects of the present disclosure can include one or more of the advantageous features discussed herein. In other words, while one or more aspects may be discussed as having certain advantageous features, one or more of such features may also be used in accordance with the various aspects of the disclosure discussed herein. In similar fashion, while exemplary aspects may be discussed below as device, system, or method embodiments such exemplary embodiments can be implemented in various devices, systems, and methods.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] So that the manner in which the above-recited features of the present disclosure can be understood in detail, a more particular description, briefly summarized above, may be had by reference to aspects, some of which are illustrated in the appended drawings. The appended drawings illustrate only certain typical aspects of this disclosure, however, and are therefore not to be considered limiting of its scope, for the description may admit to other equally effective aspects.

[0021] FIG. 1 is a block diagram conceptually illustrating an example wireless communication system, in accordance with certain aspects of the present disclosure.

[0022] FIG. 2 is a block diagram illustrating an example logical architecture of a distributed RAN, in accordance with certain aspects of the present disclosure.

[0023] FIG. 3 is a diagram illustrating an example physical architecture of a distributed RAN, in accordance with certain aspects of the present disclosure.

[0024] FIG. 4 is a block diagram conceptually illustrating a design of an example base station (BS) and user equipment (UE), in accordance with certain aspects of the present disclosure.

[0025] FIG. 5 is a diagram showing examples for implementing a communication protocol stack, in accordance with certain aspects of the present disclosure.

[0026] FIG. 6 illustrates an example of a downlink (DL)-centric subframe, in accordance with certain aspects of the present disclosure.

[0027] FIG. 7 illustrates an example of an uplink (UL)-centric subframe, in accordance with certain aspects of the present disclosure.

[0028] FIG. 8 is a graphical representation of an example low-density parity-check (LDPC) code, in accordance with certain aspects of the present disclosure.

[0029] FIG. 8A is a matrix representation of the example LDPC code of FIG. 8, in accordance with certain aspects of the present disclosure.

[0030] FIG. 9 is a graphical representation of liftings of the LDPC code of FIG. 8, in accordance with certain aspects of the present disclosure.

[0031] FIG. 10 is an integer representation of a matrix for a quasi-cyclic 802.11 LDPC code according to some aspects.

[0032] FIG. 11 is a simplified block diagram illustrating an example encoder, in accordance with certain aspects of the present disclosure.

[0033] FIG. 12 is a simplified block diagram illustrating an example decoder, in accordance with certain aspects of the present disclosure.

[0034] FIG. 13 is a flow diagram illustrating example operations for generating lifted LDPC codes from a base parity check matrix (PCM) for wireless communications by a transmitting device, in accordance with certain aspects of the present disclosure.

[0035] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. Elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

DETAILED DESCRIPTION

[0036] Aspects of the present disclosure provide apparatus, methods, processing systems, and computer program products for encoding (and/or decoding) for new radio (NR) access technology (e.g., 5G radio access). NR may refer to radios configured to operate according to a new air interface or fixed transport layer. NR may include support for enhanced mobile broadband (eMBB) service targeting wide bandwidth (e.g., 80 MHz and beyond), millimeter wave (mmW) service targeting high carrier frequency (e.g., 60 GHz), massive machine type communications (mMTC) service targeting non-backward compatible MTC techniques, and/or mission critical (MiCr) service targeting ultra-reliable low-latency communications (URLLC) service. These services may include latency and reliability requirements for a variety of uses, timing requirements, and other design considerations. NR may use low-density parity-check (LDPC) coding and/or polar codes.

[0037] Aspects of the present disclosure provide techniques and apparatus for compactly describing LDPC code structure. In aspects, a single base graph or parity check matrix (PCM) can be stored for a set of lifting sizes (sometimes referred to as a family of liftings or a family of lifted LDPC codes). The PCM may correspond to one of the liftings for the set of liftings (e.g., the smallest or largest lifting) and the other members of the family can be generated based on a stored PCM using an operation (e.g., such as a floor operation or a modulo operation). In aspects, the same PCM can be used for members of the family of codes. In aspects, PCM for different families of codes can be generated based on the lifting values associated with one family.

[0038] Various aspects of the disclosure are described more fully hereinafter with reference to the accompanying drawings. This disclosure may, however, be embodied in many different forms and should not be construed as limited to any specific structure or

function presented throughout this disclosure. Rather, these aspects are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the disclosure to those skilled in the art. Based on the teachings herein one skilled in the art should appreciate that the scope of the disclosure is intended to cover any aspect of the disclosure disclosed herein, whether implemented independently of or combined with any other aspect of the disclosure. For example, an apparatus may be implemented or a method may be practiced using any number of the aspects set forth herein. In addition, the scope of the disclosure is intended to cover such an apparatus or method which is practiced using other structure, functionality, or structure and functionality in addition to or other than the various aspects of the disclosure set forth herein. It should be understood that any aspect of the disclosure disclosed herein may be embodied by one or more elements of a claim. The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects.

[0039] Although particular aspects are described herein, many variations and permutations of these aspects fall within the scope of the disclosure. Although some benefits and advantages of the preferred aspects are mentioned, the scope of the disclosure is not intended to be limited to particular benefits, uses, or objectives. Rather, aspects of the disclosure are intended to be broadly applicable to different wireless technologies, system configurations, networks, and transmission protocols, some of which are illustrated by way of example in the figures and in the following description of the preferred aspects. The detailed description and drawings are merely illustrative of the disclosure rather than limiting, the scope of the disclosure being defined by the appended claims and equivalents thereof.

[0040] The techniques described herein may be used for various wireless communication networks such as Code Division Multiple Access (CDMA) networks, Time Division Multiple Access (TDMA) networks, Frequency Division Multiple Access (FDMA) networks, Orthogonal FDMA (OFDMA) networks, Single-Carrier FDMA (SC-FDMA) networks, etc. The terms “networks” and “systems” are often used interchangeably. A CDMA network may implement a radio technology such as Universal Terrestrial Radio Access (UTRA), CDMA2000, etc. UTRA includes Wideband-CDMA (W-CDMA) and Low Chip Rate (LCR). CDMA2000 covers IS-2000, IS-95, and IS-856

standards. A TDMA network may implement a radio technology such as Global System for Mobile Communications (GSM). An OFDMA network may implement a radio technology such as Evolved UTRA (E-UTRA), IEEE 802.11, IEEE 802.16, IEEE 802.20, Flash-OFDM®, etc. UTRA, E-UTRA, and GSM are part of Universal Mobile Telecommunication System (UMTS). 3GPP LTE and LTE-Advanced (LTE-A) are releases of UMTS that use E-UTRA. UTRA, E-UTRA, UMTS, LTE, LTE-A and GSM are described in documents from an organization named “3rd Generation Partnership Project” (3GPP). . CDMA2000 is described in documents from an organization named “3rd Generation Partnership Project 2” (3GPP2). NR is an emerging wireless communications technology under development in conjunction with the 5G Technology Forum (5GTF). These communications networks are merely listed as examples of networks in which the techniques described in this disclosure may be applied; however, this disclosure is not limited to the above-described communications network.

[0041] For clarity, while aspects may be described herein using terminology commonly associated with 3G and/or 4G or LTE wireless technologies, aspects of the present disclosure can be applied in other generation-based communication systems, such as 5G and later, including NR technologies.

AN EXAMPLE WIRELESS COMMUNICATION SYSTEM

[0042] FIG. 1 illustrates an example communications network 100 in which aspects of the present disclosure may be performed. Wireless communications network 100 may be a new radio (NR) or 5G network. Wireless communications network 100 may include a transmitting device such as a user equipment (UE) 120 or a base station (BS) 110. The transmitting device may perform encoding according to aspects described herein using lifted LDPC codes that may be compactly described (e.g., determined/generated/stored), and a receiving device (e.g., a UE 120 or a BS 110) can perform corresponding decoding operations. For example, the transmitting device can select at least one lifting size value for generating a group of lifted LDPC codes comprising copies of a base LDPC code defined by a base matrix having a first number of base variable nodes and a second number of base check nodes. The lifting size value is selected from a range of values. The transmitting device can generate the base matrix based on a lifting value of a set of lifting values associated with the selected lifting size value and generate a matrix for a different lifting size value in the group based on the base matrix.

[0043] As illustrated in FIG. 1, wireless communications network 100 may include a number of BSs 110 and other network entities. A BS may be a station that communicates with UEs. Each BS 110 may provide communication coverage for a particular geographic area. In 3GPP, the term “cell” can refer to a coverage area of a Node B and/or a Node B subsystem serving this coverage area, depending on the context in which the term is used. In NR systems, the term “cell” and gNB, Node B, 5G NB, AP, NR BS, NR BS, TRP, etc., may be interchangeable. In some examples, a cell may not necessarily be stationary, and the geographic area of the cell may move according to the location of a mobile BS. In some examples, the BSs may be interconnected to one another and/or to one or more other BSs or network nodes (not shown) in wireless communications network 100 through various types of backhaul interfaces such as a direct physical connection, a virtual network, or the like using any suitable transport network.

[0044] In general, any number of wireless networks may be deployed in a given geographic area. Each wireless network may support a particular radio access technology (RAT) and may operate on one or more frequencies. A RAT may also be referred to as a radio technology, an air interface, etc. A frequency may also be referred to as a carrier, a frequency channel, etc. Each frequency may support a single RAT in a given geographic area in order to avoid interference between wireless networks of different RATs. In some cases, NR or 5G RAT networks may be deployed.

[0045] A BS may provide communication coverage for a macro cell, a pico cell, a femto cell, and/or other types of cell. A macro cell may cover a relatively large geographic area (e.g., several kilometers in radius) and may allow unrestricted access by UEs with service subscription. A pico cell may cover a relatively small geographic area and may allow unrestricted access by UEs with service subscription. A femto cell may cover a relatively small geographic area (e.g., a home) and may allow restricted access by UEs having association with the femto cell (e.g., UEs in a Closed Subscriber Group (CSG), UEs for users in the home, etc.). A BS for a macro cell may be referred to as a macro BS. A BS for a pico cell may be referred to as a pico BS. A BS for a femto cell may be referred to as a femto BS or a home BS. In the example shown in FIG. 1, BS 110a, BS 110b, and BS 110c may be macro BSs for the macro cell 102a, macro cell 102b, and macro cell 102c, respectively. BS 110x may be a pico BS for pico cell 102x. BS

110y and BS 110z may be femto BS for the femto cell 102y and femto cell 102z, respectively. A BS may support one or multiple (e.g., three) cells.

[0046] Wireless communications network 100 may also include relay stations. A relay station is a station that receives a transmission of data and/or other information from an upstream station (e.g., a BS 110 or a UE 120) and sends a transmission of the data and/or other information to a downstream station (e.g., a UE 120 or a BS 110). A relay station may also be a UE that relays transmissions for other UEs. In the example shown in FIG. 1, relay station 110r may communicate with BS 110a and UE 120r in order to facilitate communication between BS 110a and UE 120r. A relay station may also be referred to as a relay, a relay eNB, etc.

[0047] Wireless communications network 100 may be a heterogeneous network that includes BSs of different types, for example, macro BS, pico BS, femto BS, relays, etc. These different types of BSs may have different transmit power levels, different coverage areas, and different impact on interference in the wireless communications network 100. For example, a macro BS may have a high transmit power level (e.g., 20 Watts) whereas pico BS, femto BS, and relays may have a lower transmit power level (e.g., 1 Watt).

[0048] Wireless communications network 100 may support synchronous or asynchronous operation. For synchronous operation, the BSs may have similar frame timing, and transmissions from different BSs may be approximately aligned in time. For asynchronous operation, the BSs may have different frame timing, and transmissions from different BSs may not be aligned in time. The techniques described herein may be used for both synchronous and asynchronous operation.

[0049] Network controller 130 may couple to a set of BSs and provide coordination and control for these BSs. Network controller 130 may communicate with BSs 110 via a backhaul. BSs 110 may also communicate with one another, e.g., directly or indirectly via wireless or wireline backhaul.

[0050] UEs 120 (e.g., UE 120x, UE 120y, etc.) may be dispersed throughout wireless communications network 100, and each UE may be stationary or mobile. A UE may also be referred to as a mobile station, a terminal, an access terminal, a subscriber unit, a station, a Customer Premises Equipment (CPE), a cellular phone, a smart phone, a personal digital assistant (PDA), a wireless modem, a wireless communication device, a

handheld device, a laptop computer, a cordless phone, a wireless local loop (WLL) station, a tablet, a camera, a gaming device, a netbook, a smartbook, an ultrabook, a medical device or medical equipment, a biometric sensor/device, a wearable device such as a smart watch, smart clothing, smart glasses, a smart wrist band, smart jewelry (e.g., a smart ring, a smart bracelet, etc.), an entertainment device (e.g., a music device, a video device, a satellite radio, etc.), a vehicular component or sensor, a smart meter/sensor, industrial manufacturing equipment, a global positioning system device, or any other suitable device that is configured to communicate via a wireless or wired medium. Some UEs may be considered evolved or machine-type communication (MTC) devices or evolved MTC (eMTC) devices. MTC and eMTC UEs include, for example, robots, drones, remote devices, sensors, meters, monitors, location tags, etc., that may communicate with a BS, another device (e.g., remote device), or some other entity. A wireless node may provide, for example, connectivity for or to a network (e.g., a wide area network such as Internet or a cellular network) via a wired or wireless communication link. Some UEs may be considered Internet-of-Things (IoT) devices.

[0051] In FIG. 1, a solid line with double arrows indicates desired transmissions between a UE and a serving BS, which is a BS designated to serve the UE on the downlink and/or uplink. A finely dashed line with double arrows indicates interfering transmissions between a UE and a BS.

[0052] Certain wireless networks (e.g., LTE) utilize orthogonal frequency division multiplexing (OFDM) on the downlink and single-carrier frequency division multiplexing (SC-FDM) on the uplink. OFDM and SC-FDM partition the system bandwidth into multiple (K) orthogonal subcarriers, which are also commonly referred to as tones, bins, etc. Each subcarrier may be modulated with data. In general, modulation symbols are sent in the frequency domain with OFDM and in the time domain with SC-FDM. The spacing between adjacent subcarriers may be fixed, and the total number of subcarriers (K) may be dependent on the system bandwidth. For example, the spacing of the subcarriers may be 15 kHz and the minimum resource allocation (called a “resource block” (RB)) may be 12 subcarriers (i.e., 180 kHz). Consequently, the nominal Fast Fourier Transform (FFT) size may be equal to 128, 256, 512, 1024 or 2048 for system bandwidth of 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, or 20 MHz, respectively. The system bandwidth may also be partitioned into subbands. For example, a subband may cover

1.08 MHz (i.e., 6 RBs), and there may be 1, 2, 4, 8 or 16 subbands for system bandwidth of 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, or 20 MHz, respectively.

[0053] NR may utilize OFDM with a CP on uplink and downlink and include support for half-duplex operation using TDD. A single component carrier bandwidth of 100 MHz may be supported. NR RBs may span 12 subcarriers with a subcarrier bandwidth of 75 kHz over a 0.1 ms duration. Each radio frame may consist of 50 subframes with a length of 10 ms. Consequently, each subframe may have a length of 0.2 ms. Each subframe may indicate a link direction (i.e., downlink or uplink) for data transmission and the link direction for each subframe may be dynamically switched. Each subframe may include DL/UL data as well as DL/UL control data. UL and DL subframes for NR may be as described in more detail below with respect to FIGs. 6 and 7. Beamforming may be supported and beam direction may be dynamically configured. MIMO transmissions with precoding may also be supported. MIMO configurations in the DL may support up to 8 transmit antennas with multi-layer DL transmissions up to 8 streams and up to 2 streams per UE. Multi-layer transmissions with up to 2 streams per UE may be supported. Aggregation of multiple cells may be supported with up to 8 serving cells. Alternatively, NR may support a different air interface, other than an OFDM-based.

[0054] In some examples, access to the air interface may be scheduled. For example, a scheduling entity (e.g., a BS 110 or UE 120) allocates resources for communication among some or all devices and equipment within its service area or cell. Within the present disclosure, as discussed further below, the scheduling entity may be responsible for scheduling, assigning, reconfiguring, and releasing resources for one or more subordinate entities. That is, for scheduled communication, subordinate entities utilize resources allocated by the scheduling entity. BSs are not the only entities that may function as a scheduling entity. That is, in some examples, a UE may function as a scheduling entity, scheduling resources for one or more subordinate entities (e.g., one or more other UEs). In this example, the UE is functioning as a scheduling entity, and other UEs utilize resources scheduled by the UE for wireless communication. A UE may function as a scheduling entity in a peer-to-peer (P2P) network, and/or in a mesh network. In a mesh network example, UEs may optionally communicate directly with one another in addition to communicating with the scheduling entity.

[0055] Thus, in a wireless communication network with a scheduled access to time-frequency resources and having a cellular configuration, a P2P configuration, and a mesh configuration, a scheduling entity and one or more subordinate entities may communicate utilizing the scheduled resources.

[0056] The NR radio access network (RAN) may include one or more central units (CU) and distributed units (DUs). A NR BS (e.g., a gNB, a 5G NB, a NB, a 5G NB, a TRP, an AP) may correspond to one or multiple BSs. NR cells can be configured as access cells (ACells) or data only cells (DCells). DCells may be cells used for carrier aggregation or dual connectivity, but not used for initial access, cell selection/reselection, or handover.

[0057] FIG. 2 illustrates an example logical architecture of a distributed RAN 200, which may be implemented in wireless communications system 100 illustrated in FIG. 1. 5G access node (AN) 206 may include access node controller (ANC) 202. ANC 202 may be a CU of distributed RAN 200. A backhaul interface to next generation core network (NG-CN) 204 may terminate at ANC 202. A backhaul interface to neighboring next generation access nodes (NG-ANs) may terminate at ANC 202. ANC 202 may include one or more TRPs 208.

[0058] TRPs 208 comprise DUs. TRPs 208 may be connected to one ANC (ANC 202) or more than one ANC (not illustrated). For example, for RAN sharing, radio as a service (RaaS), and service specific AND deployments, the TRP may be connected to more than one ANC 202. A TRP 208 may include one or more antenna ports. TRPs 208 may be configured to individually (e.g., dynamic selection) or jointly (e.g., joint transmission) serve traffic to a UE (e.g., a UE 120).

[0059] Example logical architecture of the distributed RAN 200 may be used to illustrate fronthaul definition. The logical architecture may support fronthauling solutions across different deployment types. For example, the logical architecture may be based on transmit network capabilities (e.g., bandwidth, latency, and/or jitter). The logical architecture may share features and/or components with LTE. NG-AN 210 may support dual connectivity with NR. NG-AN 210 may share a common fronthaul for LTE and NR. The logical architecture may enable cooperation between and among TRPs 208. For

example, cooperation may be pre-configured within a TRP 208 and/or across TRPs 208 via ANC 202. There may be no inter-TRP interface.

[0060] The logical architecture for distributed RAN 200 may include a dynamic configuration of split logical functions. As will be described in more detail with reference to FIG. 5, the Radio Resource Control (RRC) layer, Packet Data Convergence Protocol (PDCP) layer, Radio Link Control (RLC) layer, Medium Access Control (MAC) layer, and a Physical (PHY) layers may be placed at the DU (e.g., a TRP 208) or the CU (e.g., ANC 202).

[0061] FIG. 3 illustrates an example physical architecture of a distributed RAN 300, according to aspects of the present disclosure. As shown in FIG. 3, distributed RAN 300 includes centralized core network unit (C-CU) 302, centralized RAN unit (C-RU) 304, and DU 306.

[0062] C-CU 302 may host core network functions. C-CU 302 may be centrally deployed. C-CU 302 functionality may be offloaded (e.g., to advanced wireless services (AWS)), in an effort to handle peak capacity. C-RU 304 may host one or more ANC functions. Optionally, C-RU 304 may host core network functions locally. C-RU 304 may have a distributed deployment. C-RU 304 may be located near an edge the network. DU 306 may host one or more TRPs (edge node (EN), an edge unit (EU), a radio head (RH), a smart radio head (SRH), or the like). DU 306 may be located at edges of the network with radio frequency (RF) functionality.

[0063] FIG. 4 illustrates example components of the BS 110 and the UE 120 illustrated in FIG. 1, which may be used to implement aspects of the present disclosure for high performance, flexible, and compact LDPC coding. One or more of the components of BS 110 and UE 120 illustrated in FIG. 4 may be used to practice aspects of the present disclosure. For example, antenna(s) 452a-454r, Demodulator(s)/Modulator(s) 454a-454r, TX MIMO processor 466, Receive Processor 458, Transmit Processor 464, and/or Controller/Processor 480 of UE 120 and/or antenna(s) 434a-434t, Demodulator(s)/Modulator(s) 432a-434t, TX MIMO Processors 430, Transmit Processor 420, Receive Processor 438, and/or Controller/Processor 440 of BS 110 may be used to perform the operations 1300-1600 described herein and illustrated with reference to FIGs. 13-16, respectively.

[0064] For a restricted association scenario, BS 110 may be macro BS 110c in FIG. 1, and UE 120 may be UE 120y. BS 110 may also be a BS of some other type. BS 110 may be equipped with antennas 434a through 434t and UE 120 may be equipped with antennas 452a through 452r.

[0065] At BS 110, transmit processor 420 may receive data from data source 412 and control information from controller/processor 440. The control information may be for the Physical Broadcast Channel (PBCH), Physical Control Format Indicator Channel (PCFICH), Physical Hybrid ARQ Indicator Channel (PHICH), Physical Downlink Control Channel (PDCCH), or other control channel or signal. The data may be for the Physical Downlink Shared Channel (PDSCH), or other data channel or signal. Transmit processor 420 may process (e.g., encode and symbol map) the data and control information to obtain data symbols and control symbols, respectively. For example, transmit processor 420 may encode the information bits using LPDC code designs discussed in greater detail below. Transmit processor 420 may also generate reference symbols, for example, for the primary synchronization signal (PSS), secondary synchronization signal (SSS), and cell-specific reference signal (CRS). Transmit (TX) multiple-input multiple-output (MIMO) processor 430 may perform spatial processing (e.g., precoding) on the data symbols, the control symbols, and/or the reference symbols, if applicable, and may provide output symbol streams to the modulators (MODs) 432a through 432t. Each modulator 432 may process a respective output symbol stream (e.g., for OFDM, etc.) to obtain an output sample stream. Each modulator 432 may further process (e.g., convert to analog, amplify, filter, and upconvert) the output sample stream to obtain a downlink signal. Downlink signals from modulators 432a through 432t may be transmitted via antennas 434a through 434t, respectively.

[0066] At UE 120, antennas 452a through 452r may receive the downlink signals from BS 110 and may provide received signals to the demodulators (DEMODs) 454a through 454r, respectively. Each demodulator 454 may condition (e.g., filter, amplify, downconvert, and digitize) a respective received signal to obtain input samples. Each demodulator 454 may further process the input samples (e.g., for OFDM, etc.) to obtain received symbols. MIMO detector 456 may obtain received symbols from all the demodulators 454a through 454r, perform MIMO detection on the received symbols if applicable, and provide detected symbols. Receive processor 458 may process (e.g.,

demodulate, deinterleave, and decode) the detected symbols, provide decoded data for UE 120 to a data sink 460, and provide decoded control information to controller/processor 480.

[0067] On the uplink, at UE 120, transmit processor 464 may receive and process data (e.g., for the Physical Uplink Shared Channel (PUSCH) or other data channel or signal) from data source 462 and control information (e.g., for the Physical Uplink Control Channel (PUCCH) or other control channel or signal) from controller/processor 480. Transmit processor 464 may also generate reference symbols for a reference signal. The symbols from transmit processor 464 may be precoded by TX MIMO processor 466 if applicable, further processed by demodulators 454a through 454r (e.g., for SC-FDM, etc.), and transmitted to BS 110. At BS 110, the uplink signals from the UE 120 may be received by antennas 434, processed by modulators 432, detected by MIMO detector 436 if applicable, and further processed by receive processor 438 to obtain decoded data and control information sent by UE 120. Receive processor 438 may provide the decoded data to data sink 439 and the decoded control information to controller/processor 440.

[0068] Memory 442 may store data and program codes for BS 110 and memory 482 may store data and program codes for UE 120. Scheduler 444 may schedule UEs for data transmission on the downlink and/or uplink.

[0069] FIG. 5 illustrates a diagram 500 showing examples for implementing a communications protocol stack per aspects of the present disclosure. The illustrated communications protocol stacks may be implemented by devices operating in a 5G system (e.g., a system that supports uplink-based mobility). Diagram 500 illustrates a communications protocol stack including RRC layer 510, PDCP layer 515, RLC layer 520, MAC layer 525, and PHY layer 530. In an example, the layers of a protocol stack may be implemented as separate modules of software, portions of a processor or ASIC, portions of non-collocated devices connected by a communications link, or various combinations thereof. Collocated and non-collocated implementations may be used, for example, in a protocol stack for a network access device (e.g., ANs, CUs, and/or DUs) or a UE.

[0070] A first option 505-a shows a split implementation of a protocol stack, in which implementation of the protocol stack is split between a centralized network access device

(e.g., ANC 202) and distributed network access device (e.g., DU 208). In the first option 505-a, RRC layer 510 and PDCP layer 515 may be implemented by the CU, and RLC layer 520, MAC layer 525, and PHY layer 530 may be implemented by the DU. In various examples, the CU and the DU may be collocated or non-collocated. The first option 505-a may be useful in a macro cell, micro cell, or pico cell deployment.

[0071] A second option 505-b shows a unified implementation of a protocol stack, in which the protocol stack is implemented in a single network access device (e.g., access node (AN), NR BS, a NR NBa network node (NN), TRP, gNB, etc.). In the second option, RRC layer 510, PDCP layer 515, RLC layer 520, MAC layer 525, and PHY layer 530 may each be implemented by the AN. The second option 505-b may be useful in a femto cell deployment.

[0072] Regardless of whether a network access device implements part or all of a protocol stack, a UE may implement the entire protocol stack (e.g., RRC layer 510, PDCP layer 515, RLC layer 520, MAC layer 525, and PHY layer 530).

[0073] FIG. 6 is a diagram showing an example of a DL-centric subframe 600. The DL-centric subframe 600 may include control portion 602. Control portion 602 may exist in the initial or beginning portion of DL-centric subframe 600. Control portion 602 may include various scheduling information and/or control information corresponding to various portions of DL-centric subframe 600. In some configurations, control portion 602 may be a physical DL control channel (PDCCH), as shown in FIG. 6. DL-centric subframe 600 may also include DL data portion 604. DL data portion 604 may be referred to as the payload of DL-centric subframe 600. DL data portion 604 may include the communication resources utilized to communicate DL data from the scheduling entity (e.g., UE or BS) to the subordinate entity (e.g., UE). In some configurations, DL data portion 604 may be a physical DL shared channel (PDSCH).

[0074] DL-centric subframe 600 may also include common UL portion 606. Common UL portion 606 may be referred to as an UL burst, a common UL burst, and/or various other suitable terms. Common UL portion 606 may include feedback information corresponding to various other portions of DL-centric subframe 600. For example, common UL portion 606 may include feedback information corresponding to control portion 602. Non-limiting examples of feedback information may include an

acknowledgment (ACK) signal, a negative acknowledgment (NACK) signal, a HARQ indicator, and/or various other suitable types of information. Common UL portion 606 may additionally or alternatively include information, such as information pertaining to random access channel (RACH) procedures, scheduling requests (SRs), and various other suitable types of information. As illustrated in FIG. 6, the end of DL data portion 604 may be separated in time from the beginning of common UL portion 606. This time separation may be referred to as a gap, a guard period, a guard interval, and/or various other suitable terms. This separation provides time for the switchover from DL communication (e.g., reception operation by the subordinate entity (e.g., UE)) to UL communication (e.g., transmission by the subordinate entity (e.g., UE)). The foregoing is merely one example of a DL-centric subframe and alternative structures having similar features may exist without necessarily deviating from the aspects described herein.

[0075] FIG. 7 is a diagram showing an example of an UL-centric subframe 700. UL-centric subframe 700 may include control portion 702. Control portion 702 may exist in the initial or beginning portion of UL-centric subframe 700. Control portion 702 in FIG. 7 may be similar to control portion 602 described above with reference to FIG. 6. UL-centric subframe 700 may also include UL data portion 704. UL data portion 704 may be referred to as the payload of UL-centric subframe 700. UL data portion 704 may refer to the communication resources utilized to communicate UL data from the subordinate entity (e.g., UE) to the scheduling entity (e.g., UE or BS). In some configurations, control portion 702 may be a PDCCH.

[0076] As illustrated in FIG. 7, the end of control portion 702 may be separated in time from the beginning of UL data portion 704. This time separation may be referred to as a gap, guard period, guard interval, and/or various other suitable terms. This separation provides time for the switchover from DL communication (e.g., reception operation by the scheduling entity) to UL communication (e.g., transmission by the scheduling entity). UL-centric subframe 700 may also include common UL portion 706. Common UL portion 706 in FIG. 7 may be similar to the common UL portion 606 described above with reference to FIG. 6. Common UL portion 706 may additionally or alternatively include information pertaining to channel quality indicator (CQI), sounding reference signals (SRSs), and various other suitable types of information. The foregoing is merely one

example of an UL-centric subframe and alternative structures having similar features may exist without necessarily deviating from the aspects described herein.

[0077] In some circumstances, two or more subordinate entities (e.g., UEs) may communicate with each other using sidelink signals. Real-world applications of such sidelink communications may include public safety, proximity services, UE-to-network relaying, vehicle-to-vehicle (V2V) communications, Internet-of-Everything (IoE) communications, IoT communications, mission-critical mesh, and/or various other suitable applications. Generally, a sidelink signal may refer to a signal communicated from one subordinate entity (e.g., UE1) to another subordinate entity (e.g., UE2) without relaying that communication through the scheduling entity (e.g., UE or BS), even though the scheduling entity may be utilized for scheduling and/or control purposes. In some examples, the sidelink signals may be communicated using a licensed spectrum (unlike wireless local area networks (WLAN), which typically use an unlicensed spectrum).

[0078] A UE may operate in various radio resource configurations, including a configuration associated with transmitting pilots using a dedicated set of resources (e.g., a radio resource control (RRC) dedicated state, etc.) or a configuration associated with transmitting pilots using a common set of resources (e.g., an RRC common state, etc.). When operating in the RRC dedicated state, the UE may select a dedicated set of resources for transmitting a pilot signal to a network. When operating in the RRC common state, the UE may select a common set of resources for transmitting a pilot signal to the network. In either case, a pilot signal transmitted by the UE may be received by one or more network access devices, such as an AN, or a DU, or portions thereof. Each receiving network access device may be configured to receive and measure pilot signals transmitted on the common set of resources, and also receive and measure pilot signals transmitted on dedicated sets of resources allocated to the UEs for which the network access device is a member of a monitoring set of network access devices for the UE. One or more of the receiving network access devices, or a CU to which receiving network access device(s) transmit the measurements of the pilot signals, may use the measurements to identify serving cells for the UEs, or to initiate a change of serving cell for one or more of the UEs.

EXAMPLE ERROR CORRECTION CODING

[0079] Many communications systems use error-correcting codes. Error correcting

codes generally compensate for the intrinsic unreliability of information transfer (e.g., over the air medium) in these systems by introducing redundancy into the data stream. Low-density parity-check (LDPC) codes are one type of error correcting codes which use an iterative coding system. Gallager codes are an example of “regular” LDPC codes. Regular LDPC codes are linear block codes in which most of the elements of its parity check matrix H are ‘0’.

[0080] LDPC codes can be represented by bipartite graphs (often referred to as “Tanner graphs”). In a bipartite graph, a set of variable nodes corresponds to bits of a code word (e.g., information bits or systematic bits), and a set of check nodes correspond to a set of parity-check constraints that define the code. Edges in the graph connect variable nodes to check nodes. Thus, the nodes of the graph are separated into two distinctive sets and with edges connecting nodes of two different types, variable and check.

[0081] Graphs as used in LDPC coding may be characterized in a variety of manners. A lifted code is created by copying a bipartite base graph (G) (or a protograph), a number of times, Z . The number of times is referred to herein as the lifting, lifting size, or lifting size value. A variable node and a check node are considered “neighbors” if they are connected by an “edge” (i.e., the line connecting the variable node and the check node) in the graph. In addition, for each edge (e) of the bipartite base graph (G), a permutation (generally an integer value associated with the edge permutation that is represented by k and referred to as the lifting value) is applied to the Z copies of edge (e) to interconnect the Z copies of G . A bit sequence having a one-to-one association with the variable node sequence is a valid code word if and only if, for each check node, the bits associated with all neighboring variable nodes sum to 0 modulo 2 (i.e., they include an even number of 1’s). The resulting LDPC code may be quasi-cyclic (QC) if the permutations (liftings values) used are cyclic.

[0082] FIGs. 8-8A show graphical and matrix representations, respectively, of an example LDPC code, in accordance with certain aspects of the present disclosure. For example, FIG. 8 shows a bipartite graph 800 representing an example LDPC code. Bipartite graph 800 includes a set of five variable nodes 810 (represented by circles) connected to four check nodes 820 (represented by squares). Edges in bipartite graph 800 connect variable nodes 810 to check nodes 820 (the edges are represented by the lines

connecting variable nodes 810 to check nodes 820). Bipartite graph 800 consists of $|V| = 5$ variable nodes and $|C| = 4$ check nodes, connected by $|E| = 12$ edges.

[0083] Bipartite graph 800 may be represented by a simplified adjacency matrix, which may also be known as a parity check matrix (PCM). FIG. 8A shows a matrix representation 800A of bipartite graph 800. Matrix representation 800A includes a PCM H and a code word vector x , where x_1 - x_5 represent bits of the code word x . H is used for determining whether a received signal was normally decoded. H has C rows corresponding to j check nodes and V columns corresponding to i variable nodes (i.e., a demodulated symbol), where the rows represent the equations and the columns represents the bits of the code word. In FIG. 8A, matrix H has four rows and five columns corresponding to four check nodes and five variable nodes, respectively. If a j -th check node is connected to an i -th variable node by an edge (i.e., the two nodes are neighbors), then there is a 1 in the i -th column and in the j -th row of the parity check matrix H . That is, the intersection of an i -th row and a j -th column contains a "1" where an edge joins the corresponding vertices and a "0" where there is no edge. The code word vector x represents a valid code word if and only if $H_x = 0$, for example, if for each constraint node, the bits neighboring the constraint, via their association with variable nodes, sum to 0 modulo 2 (i.e., they comprise an even number of 1's). Thus, if the code word is received correctly, then $H_x = 0 \pmod{2}$. When the product of a coded received signal and the PCM H becomes '0', this signifies that no error has occurred.

[0084] The number of demodulated symbols or variable nodes is the LDPC code length. The number of non-zero elements in a row (column) is defined as the row (column) weight $d(c)d(v)$. The degree of a node refers to the number of edges connected to that node. For example, as shown in FIG. 8, the variable node 801 has three degrees of connectivity, with edges connected to check nodes 811, 812, and 813. Variable node 802 has three degrees of connectivity, with edges connected to check nodes 811, 813, and 814. Variable node 803 has two degrees of connectivity, with edges connected to check nodes 811 and 814. Variable node 804 has two degrees of connectivity, with edges connected to check nodes 812 and 814. And variable node 805 has two degrees of connectivity, with edges connected to check nodes 812 and 813. This feature is illustrated in the matrix H shown in FIG. 8A where the number of edges incident to a variable node 810 is equal to the number of 1's in the corresponding column and is called the variable node degree $d(v)$.

Similarly, the number of edges connected with a check node 820 is equal to the number of ones in a corresponding row and is called the check node degree $d(c)$. For example, as shown in FIG. 8A, the first column in the matrix H corresponds to the variable node 801 and the corresponding entries in the column (1, 1, 1, 0) indicates the edge connections to the check nodes 811, 812, and 813, while the 0 indicates that there is not an edge to check node 814. The entries in the second, third, fourth, and fourth columns of H represent the edge connections of the variable nodes 802, 803, 804, and 805, respectively, to the check nodes.

[0085] A regular graph or a regular code is one for which all variable nodes have the same degree and all constraint nodes have the same degree. On the other hand, an irregular code has constraint nodes and/or variable nodes of differing degrees. For example, some variable nodes may be of degree 4, others of degree 3, and still others of degree 2.

[0086] “Lifting” enables LDPC codes to be implemented using parallel encoding and/or decoding implementations while also reducing the complexity typically associated with large LDPC codes. Lifting helps enable efficient parallelization of LDPC decoders while still having a relatively compact description. More specifically, lifting is a technique for generating a relatively large LDPC code from multiple copies of a smaller base code. For example, a lifted LDPC code may be generated by producing Z of parallel copies of the base graph (e.g., protograph) and then interconnecting the parallel copies through permutations of edge bundles of each copy of the base graph. The base graph defines the (macro) structure of the code and consists of a number (K) of information bit columns and a number (N) of code bit columns. Lifting the base graph a number of liftings Z results in a final block length of KZ . Thus, a larger graph can be obtained by a “copy and permute” operation where multiple copies of the base graph are made and connected to form a single lifted graph. For the multiple copies, like edges are a set of copies of single base edge, are permuted and connected to form a connected graph Z times larger than the base graph.

[0087] FIG. 9 is a bipartite graph illustrating liftings of three copies of the bipartite graph 800 of FIG. 8. Three copies may be interconnected by permuting like edges among the copies. If the permutations are restricted to cyclic permutations, then the resulting bipartite graph 900 corresponds to a quasi-cyclic LDPC with lifting $Z = 3$. The original

graph 800 from which three copies were made is referred to herein as the base graph. To obtain graphs of different sizes, “copy and permute” operation can be applied to the base graph.

[0088] A corresponding PCM of the lifted graph can be constructed from the parity check matrix of the base graph by replacing each entry in the base parity check matrix with a $Z \times Z$ matrix. The “0” entries (those having no base edges) are replaced with the 0 matrix and the 1 entries (indicating a base edge) are replaced with a $Z \times Z$ permutation matrix. In the case of cyclic liftings, the permutations are cyclic permutations.

[0089] A cyclically lifted LDPC code can also be interpreted as a code over the ring of binary polynomials modulo $x^Z + 1$. In this interpretation, a binary polynomial, $(x) = b_0 + b_1x + b_2x^2 + \dots + b_{Z-1}x^{Z-1}$ may be associated to each variable node in the base graph. The binary vector $(b_0, b_1, b_2, \dots, b_{Z-1})$ corresponds to the bits associated to Z corresponding variable nodes in the lifted graph, that is, Z copies of a single base variable node. A cyclic permutation by k (referred to as a lifting value associated to the edges in the graph) of the binary vector is achieved by multiplying the corresponding binary polynomial by x^k where multiplication is taken modulo $x^Z + 1$. A degree d parity check in the base graph can be interpreted as a linear constraint on the neighboring binary polynomials $B_1(x), \dots, B_d(x)$, written as $x^{k_1}B_1(x) + x^{k_2}B_2(x) + \dots + x^{k_d}B_d(x) = 0$ or $x^{k_1}B_1(x) + x^{k_2}B_2(x) + \dots + x^{k_d}B_d(x) = 0$, the values, $k_1, \dots, k_d$ are the cyclic lifting values associated to the corresponding edges.

[0090] This resulting equation is equivalent to the Z parity checks in the cyclically lifted Tanner graph corresponding to the single associated parity check in the base graph. Thus, the parity check matrix for the lifted graph can be expressed using the matrix for the base graph in which 1 entries are replaced with monomials of the form x^k and 0 entries are lifted as 0, but now the 0 is interpreted as the 0 binary polynomial modulo $x^Z + 1$. Such a matrix may be written by giving the value k in place of x^k . In this case the 0 polynomial is sometimes represented as “-1” and sometimes as another character in order to distinguish it from x^0 .

[0091] Typically, a square submatrix of the parity check matrix represents the parity bits of the code. The complementary columns correspond to information bits that, at the time of encoding, are set equal to the information bits to be encoded. The encoding may be achieved by solving for the variables in the aforementioned square submatrix in order

to satisfy the parity check equations. The parity check matrix H may be partitioned into two parts M and N where M is the square portion. Thus, encoding reduces to solving $M_c = s = Nd$ where c and d comprise x . In the case of quasi-cyclic codes, or cyclically lifted codes, the above algebra can be interpreted as being over the ring of binary polynomials modulo $x^z + 1$. In the case of the 802.11 LDPC codes, which are quasi-cyclic, the encoding submatrix M has an integer representation as shown in FIG. 10.

[0092] A received LDPC code word can be decoded to produce a reconstructed version of the original code word. In the absence of errors, or in the case of correctable errors, decoding can be used to recover the original data unit that was encoded. Redundant bits may be used by decoders to detect and correct bit errors. LDPC decoder(s) generally operate by iteratively performing local calculations and passing those results by exchanging messages within the bipartite graph along the edges, and updating these messages by performing computations at the nodes based on the incoming messages. These steps may be repeated several times. For example, each variable node 810 in the graph 800 may initially be provided with a “soft bit” (e.g., representing the received bit of the code word) that indicates an estimate of the associated bit’s value as determined by observations from the communications channel. Using these soft bits the LDPC decoders may update messages by iteratively reading them, or some portion thereof, from memory and writing an updated message, or some portion thereof, back to, memory. The update operations are typically based on the parity check constraints of the corresponding LDPC code. In implementations for lifted LDPC codes, messages on like edges are often processed in parallel.

[0093] LDPC codes designed for high speed applications often use quasi-cyclic constructions with large lifting factors and relatively small base graphs to support high parallelism in encoding and decoding operations. LDPC codes with higher code rates (e.g., the ratio of the message length to the codeword length) tend to have relatively fewer parity checks. If the number of base parity checks is smaller than the degree of a variable node (e.g., the number of edges connected to a variable node), then, in the base graph, that variable node is connected to at least one of the base parity checks by two or more edges (e.g., the variable node may have a “double edge”). If the number of base parity checks is smaller than the degree of a variable node (e.g., the number of edges connected to a variable node), then, in the base graph, that variable node is connected to at least one of

the base parity checks by two or more edges. Having a base variable node and a base check node connected by two or more edges is generally undesirable for parallel hardware implementation purposes. For example, such double edges may result in multiple concurrent read and write operations to the same memory locations, which in turn may create data coherency problems. A double edge in a base LDPC code may trigger parallel reading of the same soft bit value memory location twice during a single parallel parity check update. Thus, additional circuitry is typically needed to combine the soft bit values that are written back to memory, so as to properly incorporate both updates. Eliminating double edges in the LDPC code helps to avoid this extra complexity.

[0094] LDPC code designs based on cyclic lifting can be interpreted, as codes over the ring of polynomials modulo $x^Z - 1$, where Z is the lifting size (e.g., the size of the cycle in the quasi-cyclic code). Thus encoding such codes can often be interpreted as an algebraic operation in this ring.

[0095] In the definition of standard irregular LDPC code ensembles (degree distributions) all edges in the Tanner graph representation may be statistically interchangeable. In other words, there exists a single statistical equivalence class of edges. A more detailed discussion of lifted LDPC codes may be found, for example, in the book titled, "Modern Coding Theory," published Mar. 17, 2008, by Tom Richardson and Ruediger Urbanke. For multi-edge LDPC codes, multiple equivalence classes of edges may be possible. While in the standard irregular LDPC ensemble definition, nodes in the graph (both variable and constraint) are specified by their degree, i.e., the number of edges they are connected to, in the multi-edge type setting an edge degree is a vector; it specifies the number of edges connected to the node from each edge equivalence class (type) independently. A multi-edge type ensemble is comprised of a finite number of edge types. The degree type of a constraint node is a vector of (non-negative) integers; the i -th entry of this vector records the number of sockets of the i -th type connected to such a node. This vector may be referred to as an edge degree. The degree type of a variable node has two parts although it can be viewed as a vector of (non-negative) integers. The first part relates to the received distribution and will be termed the received degree and the second part specifies the edge degree. The edge degree plays the same role as for constraint nodes. Edges are typed as they pair sockets of the same type. The constraint that sockets must pair with sockets of like type characterizes the multi-edge

type concept. In a multi-edge type description, different node types can have different received distributions (e.g., the associated bits may go through different channels).

[0096] Puncturing is the act of removing bits from a codeword to yield a shorter codeword. Thus, punctured variable nodes correspond to codeword bits that are not actually transmitted. Puncturing a variable node in an LDPC code creates a shortened code (e.g. due to the removal of a bit), while also effectively removing a check node. Specifically, for a matrix representation of an LDPC code, including bits to be punctured, where the variable node to be punctured has a degree of one (such a representation may be possible through row combining provided the code is proper), puncturing the variable node removes the associated bit from the code and effectively removes its single neighboring check node from the graph. As a result, the number of check nodes in the graph is reduced by one.

[0097] FIG. 11 is a simplified block diagram illustrating an encoder, in accordance with certain aspects of the present disclosure. FIG. 11 is a simplified block diagram 1100 illustrating a portion of radio frequency (RF) modem 1150 that may be configured to provide a signal including an encoded message for wireless transmission. In one example, convolutional encoder 1102 in a BS 110 (or a UE 120 on the reverse path) receives message 1120 for transmission. Message 1120 may contain data and/or encoded voice or other content directed to the receiving device. Encoder 1102 encodes the message using a suitable modulation and coding scheme (MCS), typically selected based on a configuration defined by BS 110 or another network entity. Encoded bitstream 1122 produced by encoder 1102 may then be selectively punctured by puncturing module 1104, which may be a separate device or component, or which may be integrated with encoder 1102. Puncturing module 1104 may determine that bitstream 1122 should be punctured prior to transmission, or transmitted without puncturing. The decision to puncture bitstream 1122 is typically made based on network conditions, network configuration, RAN defined preferences and/or for other reasons. Bitstream 1122 may be punctured according to puncture pattern 1112 and used to encode message 1120. Puncturing module 1104 provides output 1124 to mapper 1106 that generates a sequence of Tx symbols 1126 that are modulated, amplified and otherwise processed by Tx chain 1108 to produce an RF signal 1128 for transmission through antenna 1110.

[0098] Output 1124 of puncturing module 1104 may be the unpunctured bitstream

1122 or a punctured version of the bitstream 1122, according to whether modem portion 1150 is configured to puncture the bitstream 1122. In one example, parity and/or other error correction bits may be punctured in output 1124 of encoder 1102 in order to transmit message 1120 within a limited bandwidth of the RF channel. In another example, the bitstream may be punctured to reduce the power needed to transmit message 1120, to avoid interference, or for other network-related reasons. These punctured code word bits are not transmitted.

[0099] The decoders and decoding algorithms used to decode LDPC codewords operate by exchanging messages within the graph along the edges and updating these messages by performing computations at the nodes based on the incoming messages. Each variable node in the graph is initially provided with a soft bit, termed a received value, that indicates an estimate of the associated bit's value as determined by observations from, for example, the communications channel. Ideally, the estimates for separate bits are statistically independent. This ideal may be violated in practice. A received word is comprised of a collection of received values.

[0100] FIG. 12 is a simplified block diagram illustrating a decoder, in accordance with certain aspects of the present disclosure. FIG. 12 is a simplified schematic 1200 illustrating a portion of a RF modem 1250 that may be configured to receive and decode a wirelessly transmitted signal including a punctured encoded message. The punctured code word bits may be treated as erased. For example, the log-likelihood ratios (LLRs) of the punctured nodes may be set to 0 at initialization. De-puncturing may also include deshortening of shortened bits. These shortened bits are not included in a transmission and, at the receiver/decoder, shortened bits are treated as known bits. In various examples, modem 1250 receiving the signal may reside at the UE, at the BS, or at any other suitable apparatus or means for carrying out the described functions. Antenna 1202 provides an RF signal 1220 to a receiver. RF chain 1204 processes and demodulates RF signal 1220 and may provide a sequence of symbols 1222 to demapper 1226, which produces a bitstream 1224 representative of the encoded message.

[0101] Demapper 1206 may provide a depunctured bitstream 1224. In one example, demapper 1206 may include a depuncturing module that can be configured to insert null values at locations in the bitstream at which punctured bits were deleted by the transmitter. The depuncturing module may be used when the puncture pattern 1210 used

to produce the punctured bitstream at the transmitter is known. Puncture pattern 1210 can be used to identify LLRs 1228 that may be ignored during decoding of bitstream 1224 by convolutional decoder 1208. The LLRs may be associated with a set of depunctured bit locations in the bitstream 1224. Accordingly, decoder 1208 may produce decoded message 1226 with reduced processing overhead by ignoring the identified LLRs 1228. The LDPC decoder may include a plurality of processing elements to perform the parity check or variable node operations in parallel. For example, when processing a code word with lifting size Z , the LDPC decoder may utilize a number (Z) of processing elements to perform parity check operations on all edges of a lifted graph, concurrently.

[0102] Processing efficiency of decoder 1208 may be improved by configuring decoder 1208 to ignore LLRs 1228 that correspond to punctured bits in a message transmitted in a punctured bitstream 1222. The punctured bitstream 1222 may have been punctured according to a puncturing scheme that defines certain bits to be removed from an encoded message. In one example, certain parity or other error-correction bits may be removed. A puncturing pattern may be expressed in a puncturing matrix or table that identifies the location of bits to be punctured in each message. A puncturing scheme may be selected to reduce processing overhead used to decode the message 1226 while maintaining compliance with data rates on the communication channel and/or with transmission power limitations set by the network. A resultant punctured bitstream typically exhibits the error-correcting characteristics of a high rate error-correction code, but with less redundancy. Accordingly, puncturing may be effectively employed to reduce processing overhead at the decoder 1208 in the receiver when channel conditions produce a relatively high signal to noise ratio (SNR).

[0103] At the receiver, the same decoder used for decoding non-punctured bitstreams can typically be used for decoding punctured bitstreams, regardless of how many bits have been punctured. In conventional receivers, the LLR information is typically de-punctured before decoding is attempted by filling LLRs for punctured states or positions (de-punctured LLRs) with 0's. The decoder may disregard de-punctured LLRs that effectively carry no information based, at least in part, on which bits are punctured. The decoder may treat shortened bits as known bits (e.g., set to 0).

EXAMPLE: COMPACTLY DESCRIBED LIFTED LDPC CODE FEATURES

[0104] Certain systems (e.g., 802.11n, 802.11ad, WiMAX, ATSC, etc.) may use a multi-edge (ME) type low-density parity-check (LDPC) code structure. Multi-edge type LDPC codes may have advantages over the standard irregular LDPC codes. The ME framework may provide a framework for design of high-performance LDPC codes by using state nodes. The multi-edge type LDPC code structure may provide many more degrees of freedom than the standard irregular LDPC codes, which can be exploited to design codes with excellent performance, low encoding/decoding complexity, and/or other desirable properties. ME codes may have an accumulate chain of degree 2 parity-bits which make the code systematic and, thus, easy to encode.

[0105] ME type LDPC codes may appear in the form of protograph based LDPC codes, in which the ME type LDPC codes are formed from a base parity-check matrix (PCM). As described above, the protograph and PCMs are used to represent an (n, k) LDPC codes. The PCM defines the base structure or the edge-types in the code.

[0106] As described above, LDPC codes can be lifted by taking Z (size of the lift) copies of the base PCM and assigning random permutations (according to integer lifting values k) to each edge bundle to interconnect the Z copies and obtain the final PCM. The final PCM has a blocklength Z times the size of the base PCM. Typically, the permutation used is a cyclic permutation (e.g., using circulant matrices to obtain the final PCM). The final PCM can be represented by replacing the non-zero entries in the base PCM by integers up to the size $Z - 1$. The integer represents the cyclic shift (by that integer value) associated to the lifted bundle of edges in the lifted code structure.

[0107] In some cases, a range of blocklengths may be transmitted. In this case, different values of Z can be used for the same base graph to achieve different blocklengths (since the blocklength is equal to Z times the length of the base PCM). To obtain different code rates, different PCMs and/or different permutation (lifting values) can be used, for example, for a same lifting size Z .

[0108] As an example, in the 802.11n standard the base PCM has codeblock length equal to 24 and the lifting sizes are given by $Z = 27, 54, 81$. This gives codeblock lengths of 648, 1296 and 1944, respectively (e.g., by multiplying 24×27 , 24×54 , and 24×81 , respectively). In the example of the 802.11n standard approach, a unique PCM is defined for each code rate and each blocklength. In 802.11n, there are four code rate points, thus,

the number of defined PCMs is twelve (e.g., one PCM for each combination of the 4 code rates x 3 codeblock lengths).

[0109] In this case, when the number of blocklengths and code rate is large (e.g., as typically is the case in long term evolution (LTE)), describing (e.g., defining/generating/determining/storing) a different PCM for each pair of code rate and blocklength can lead to a large microcode to describe the PCMs (e.g., a large number of bits needed to store the different PCMs).

[0110] Accordingly, techniques for a compact description of PCMs for large numbers of blocklengths and code rates, while maintaining high performance, are desirable.

[0111] A technique for generating lifted LDPC codes (e.g., lifted ME LDPC codes) is provided herein, which lends itself to a compact description and provides finely granular blocklength scaling.

[0112] FIG. 13 illustrates example operations 1300 for wireless communication, in accordance with certain aspects of the present disclosure. Operations 1300 may be performed, for example, by a transmitting device (e.g., UE 120 or BS 110). Operations 1300 begin, at 1302, by selecting a lifting size value Z (e.g., leader) and a first set of lifting values for generating a first lifted LDPC code (e.g., a multi-edge LDPC code). The lifting size value may be selected to achieve a target blocklength or range of blocklengths. At 1304, the transmitting device generates the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base PCM having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code (an edge is a connection between a variable node and a check node). At 1306, the transmitting device determines a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values. At 1308, the transmitting device encodes a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word. At 1310, the transmitting device transmits the code word via a wireless medium.

[0113] According to certain aspects, the matrix for the different lifting size value in the group is generated based on the base matrix by performing an operation involving the

different lifting size value and an integer value associated with an edge in the base matrix (e.g., associated with a permutation of an entry in the base matrix), such as a modulo operations or a floor operation.

[0114] According to certain aspects, a plurality of lifting size values (e.g., leaders) can be selected (e.g., families of a same code rate). A PCM can be generated for each of the selected lifting size values and members in the associated groups (e.g., families or equivalent classes) can be generated based on the PCM for their respective leader. In aspects, the transmitting device can store only the generated matrices (e.g., PCMs) that are based on the maximum lifting values (from which the PCMs for the other members in the family can be generated).

[0115] Lifted LDPC codes can be described by assigning to each edge of the PCM a number (e.g., an integer) which may be less than the size of the lifting Z . That entry can be replaced with a circulant matrix obtained by cyclically shifting (e.g., to the right), the identity matrix of size $Z \times Z$, by that number. Thus, PCMs can be given by a matrix of the size of the base PCM with integer entries corresponding to the cyclic liftings.

[0116] According to certain aspects, a set of lifts can be defined as $\{a_1, a_2, \dots, a_k\} \times 2^i$, $0 \leq i \leq m$, where k is number of families or equivalent classes (e.g., associated with a same code rate), a_j is family leader (e.g., a positive integer), and m is a maximum power of 2 (e.g., the number of members in each family or granularity of blocklengths that can be obtained). The smallest power of 2 may be 0, and the largest power of 2, m , may be a large number (e.g., 10) depending on the maximum desired blocklength.

[0117] Each family $j \in [1:k]$ can be described (e.g., generated) by its leader a_j and its power of 2 up to the maximum power m . In this case, there are $(m+1)k$ lifting values. As an example, for four families or equivalent classes ($k=4$), a maximum power of 2 of six ($m=6$) can be used, where the leaders of the four families can be 16 ($a_1=16$), 20 ($a_2=20$), 24 ($a_3=24$), and 28 ($a_4=28$). In this the example, the lifting values of the four families can be defined (e.g., generated) based on power of 2 of the leader up to maximum power m (6 in this example). Thus, the possible values of the lifts for the four families in this example can be given as:

$$Z = 16 \times \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\}$$

$$Z = 20 \times \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\}$$

$$Z = 24 \times \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\}$$

$$Z = 28 \times \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\}$$

corresponding, respectively, to:

$$Z = 16, 32, 64, 128, 256, 512, 1024$$

$$Z = 20, 40, 80, 160, 320, 640, 1280$$

$$Z = 24, 48, 96, 192, 384, 768, 1536$$

$$Z = 28, 56, 112, 224, 448, 896, 1792$$

[0118] If independent lift values are used for each lift (e.g., blocklength), then $(m + 1)k$ PCMs (e.g., one PCM for each lifting value) are defined for each code rate point. In the example described above, for the four families and maximum power of 6, 28 PCMs (e.g., $(6 + 1) \times 4 = 28$) are defined.

[0119] According to certain aspects, instead of defining (e.g., obtaining/generating/determining/storing) a PCM for each lifting value, a single PCM can be used for each family, for example, only the PCM corresponding to the largest lifting in the family (e.g., the leader times the maximum power of 2). For a family j , the PCM can be generated for the largest lifting value $a_j 2^m$. The other lifting size values in the family include the leader multiplied by the other powers of 2. PCMs for the other members of the family can be obtained based on the PCM for the largest lifting. For example, edges in the PCM for the leader can be associated with a set of lifting values. PCM for the other members in the family of codes can be obtained by performing an operation involving the PCM for the largest lifting size value (e.g., based on the lifting values) and the desired lift size (e.g., the lift size of that family member). The operations may include a modulo with respect to the desired lift size, a floor operation, or other operation. For example, if an edge has a value s (e.g., an integer value) in the PCM (e.g., for the largest lifting size value) of the family j , then the corresponding integer value in the PCM for the desired lifting size value l , $l < m$, can be given by $s \bmod a_j 2^l$. In the example of the floor operation, the corresponding integer value may be found by $\text{floor}(s * (\text{desired lifting size value}) / (\text{maximum lifting size value}))$:

$$\text{floor}(x) = \lfloor s * \frac{l}{\text{max lift}} \rfloor$$

[0120] For each code rate, a PCM can be described for each family which corresponds to the PCM of the largest lift size in that family. Thus, for each code rate point, there are as many PCMs as there are families.

[0121] In the example described above, for each code rate, four PCMs may be used—one PCM for each of the maximum lifting size value of the four families 1024, 1280, 1536, and 1792, respectively. That is, the PCM described for each family is for the largest lift size. In the case of the family with the leader of 16, an edge in the base graph (e.g., the PCM for the largest lifting size value in the family of 16 is $16 \cdot 2^6 = 1024$) may be associated with an integer lifting value s of 678 (“ s ” is the value of the integer which represents the cyclic lift for that particular edge.) Values associated to all edges in the base PCM may be less than the maximum lifting size (1024 in this example). To obtain the integer value corresponding to the same edge in the graph for a different lifting size in the family, the operation $s \bmod Z$ (e.g., when $Z = 128$ the operation is $678 \bmod 128 = 38$) can be performed to generate the lifting values for the PCM for that member of the family. Thus, when lifting with the lifting size value $Z = 128$, for that edge, a circulant matrix of size 128×128 may be used, for example, which is the identity matrix (of size 128×128) shifted to the right by 38. Although in this example, a modulo operation is used, in other embodiments, as discussed above, a different operation may be used (e.g., a floor operation or other operation).

[0122] LDPC codes are often designed so that the decoding graph has few small loops, for example, due to the nature of the iterative decoders whose performance degrades in the presence of small loops. A loop (or cycle) is defined as a closed path with no repeated nodes. This implies it has an even length. Liftings can be chosen to achieve loop properties in the graph. Accordingly, it is desirable to have a scheme for generating LDPC codes for multiple liftings that all have good loop properties (e.g., few, or no, small loops). Decoding LDPC codes to find the most likely original message involves passing probability messages around the graph of an LDPC code. The decoding is usually quickly computed if the graph contains no loops. Unfortunately, strong LDPCs use loops in their graphs. As a result, the LDPC algorithm is iterated repeatedly until it is told to stop or it converges to a solution. The reason is that in a short loop the value of an incorrect bit will propagate back around to itself, effectively reinforcing its belief and resisting efforts of the

LDPC algorithm to correct it. However, this effect is diluted with longer loops, and does not affect the decoder's performance as much.

[0123] Taking the modulo with respect to any arbitrary lift size (e.g., a lift size that does not belong to the family) could lead to generating a PCM that has bad loops/sets, which can lead to error-floors and degradation in performance.

[0124] Each family j may be designed (e.g., optimized) such that taking the modulo within a family does not cause, or minimizes, the formation of bad loops. For example, the PCM for the lift size a_j can be selected (e.g., designed) such that the formation of bad loops is minimized or avoided. The lift of size $a_j/2$ can be obtained by considering the PCM obtained in the first step and then further lifted by 2 so that the number of bad loops in the second step are minimized. This continues until the PCM for the largest lift size in the family is generated. Thus, generating a family involves describing the PCM corresponding to the largest lift size in the family by multiplying its leader and its power of 2 up to a maximum number. Next, the lifting values for the remaining liftings in the family are obtained using a modulo operation.

[0125] According to certain aspects, for each code rate point, the PCM may be extended for an incremental redundancy (IR) hybrid automatic repeat request (HARQ) scheme. For each code rate point (e.g., the first transmission), the extended (IR HARQ) PCM can be described (e.g., generated) for each family corresponding to the largest lift size.

EXAMPLE: INDEPENDENT CLUSTERING SCHEME FOR EFFICIENTLY LIFTING LDPC CODES

[0126] In a wireless communication system (e.g., wireless communications system 100), a set of error correcting codes (e.g., LDPC codes) may be used, for example, for various ranges of blocklengths and/or code rates to be used. To increase efficiency in terms of implementation and compactness of description, it is desirable that the set of codes are related.

[0127] As described above with respect to FIG. 9, a base graph or parity check matrix (PCM) (having K information bits-columns and N total transmitted bit-columns) can be copied, and random permutations to each edge bundle to interconnect the copies, to provide a lifted LDPC code. Practical codes use cyclic permutations or circulant

permutation matrices to interconnect the copies of the lifted base graph, resulting in quasi-cyclic codes, which may be easier to implement in hardware. In an example, for a lifting value Z , each edge in the base PCM is associated with an integer lifting value k in the range $[0, Z-1]$. The associated integer represents the cyclic shift of the identity matrix by that integer. A table may be used for the base PCM showing entries for the bit columns and check nodes. Each entry corresponds to the circulant matrix that is the identity matrix cyclically shifted by the integer value associated with an edge between a variable node and a check node. The entry '.' May be used when there is no edge present between a base variable node and a base check node.

[0128] When the base graph is reused without alteration the code rate (given by K/N) is same for all liftings Z (corresponding to the number of liftings or copies of the base graph). Using different lifting values can provide a set of codes (e.g., a code family) to achieve a range of block lengths (given by KZ). Thus, using different lifting values for the unaltered base graph can achieve a set of codes with a similar code rate but for different block lengths. For different codes rates, different base graphs may be used.

[0129] To generate/describe a set of codes (e.g., code family) for a range of code rates and/or block lengths, one way to design the code family is to design a different base PCM for each code rate and each lift value. For example, in 802.11n there are four code rates ($1/2$, $2/3$, $3/4$, $5/6$) and three blocklengths (648, 1296, 1944) corresponding to the lift values of (27, 54, 81). There is a unique base PCM of size 24 bit-columns for each "tuple" (i.e., each pair of code rate and lift value) resulting in twelve base PCMs (e.g., for the combinations of code rate and lift value: $(1/2, 27)$, $(1/2, 54)$, $(1/2, 81)$,... $(5/6, 81)$). Thus, for large Z , the set of liftings Z and lifting values k can lead to a large description complexity.

[0130] Techniques for efficiently describing/generating the set of liftings are desirable. A set of liftings for a single parity matrix may be efficiently described as an increasing series of liftings that are closely spaced to each other in value. This allows liftings to be specified in a narrow range with a common set of bits, allowing for a compact description and good performance.

[0131] In an example, a transmitter/encoder device (e.g., such as a BS 110 or a UE 120) determines a base matrix that is associated with a cluster of lifting size values. The

transmitting device selects a lifting size value, Z , for generating a lifted LDPC code by permutations of edges in the base matrix. The lifting size values in the cluster of lifting size values are within a defined range of each other. The transmitting device generates a lifted matrix based on the base matrix and/or the selected lifting size value. The transmitting device uses the generated lifted matrix to generate the lifted LDPC code, encodes a set of information bits based on the lifted LDPC code to produce a code word, and transmits the code word over a wireless medium.

[0132] According to aspects of the present disclosure, a set of liftings Z for a single base graph or PCM, to obtain a family of LDPC codes can be described (e.g., determined/generated) using lifting values that are close to each other in value for a compact description.

[0133] The family of LDPC codes can be obtained using a base graph together with an increasing series of liftings with lifting values $Z_1, Z_2, \dots, Z_n$ which may be referred to herein as a “tower” of liftings. A cluster includes members which are within a defined range of each other. For example, members of a cluster may be within a certain ratio of each other. In some cases, the values of the members of the cluster may be within a ratio of two of each other.

[0134] One example of a cluster is the set of lifting values $\{4, 5, 6, 7\}$ having a maximum ratio of $7/4$. A tower can be obtained by applying an exponential power to an integer, such as a power of 2. Thus, a tower of clustered liftings may consist of the integers $2^j \{4, 5, 6, 7\}$ for $j=1, \dots, 7$. This gives an approximately exponentially spaced set of 28 values for Z . Put another way, this gives the tower $Z_1, Z_2, \dots, Z_{28} = 8 (2^1 \cdot 4), 10, 12, 14, \dots, 896 (2^7 \cdot 7)$. For a fixed j the four lifting values are within a factor of $7/4$ of each other and may form a cluster of lifting values. For $j=1, \dots, 7$, a tower of clustered liftings may be represented as $2^j \{4, 5, 6, 7\}$. While the present example includes a set of lifts within a factor of 2 as clustered, other factors, (e.g., 3, 4, ..., etc.) may be used. These factors need not be consecutive, but should be numerically within a defined range of each other.

[0135] According to certain aspects, for any lifting size Z in the set of clustered liftings, the associated integer lifting values k for the edge permutations may be used for any of the other liftings in the set of clustered liftings. For example, lifting values may be

designed for $Z=2^j$ that are also good for $2^j\{5, 6, 7\}$. Thus, describing (e.g., determining/generating/indicating/storing) a family of LDPC codes may be performed by identifying sets of clustered lift values (associated to edges in a base graph) that are close to each other, such as within a factor (e.g., a factor 2 or 3) of each other. In the example above, this corresponds to identifying the set of lifting values $\{4, 5, 6, 7\}$ and the other sets in the tower of liftings, $\{16, 20, 24, 28\}$, $\{32, 40, 48, 56\}$,... $\{512, 640, 768, 896\}$, which are within a factor of 2 of each other. For each clustered set of liftings, the base PCM for the smallest lift value in the cluster (e.g., $Z=8$) may be optimized. That optimized base PCM may be used for the other lift values in that cluster (e.g., $Z=10$, $Z=12$, $Z=14$). Similarly, the optimized base PCM can be determined for the other sets of clustered liftings.

[0136] Thus, liftings within a defined range of each can be specified (e.g., stored/indicated) other with a common set of bits. For example, $j+2$ bits per lifting value may be used to specify all lifts for the four stated liftings in the cluster $2^j\{4,5,6,7\}$.

[0137] These liftings may be further improved by having additional bits. For example, using $j+3$ bit to represent the lifting values k on an edge and defining the lifting by taking the $j+3$ bit value modulo Z for Z in $2^j\{4, 5, 6, 7\}$ results in a lifting for $Z=2^j*4$ given by the $j+2$ lower order bits and the higher order bit affects only the other 3 liftings. Higher order bits can similarly be used. The example presents a range of liftings within a factor of 2 of each other and all are specified using a $j+2$ (or slightly larger) bits. However, other factors may be used, so long as the factors are numerically within a defined range of each other.

[0138] Generally, optimization of lifts and graphs targets reducing the number of small loops in the Tanner graph of the LDPC code. A loop in the lifted Tanner graph corresponds with a loop in the base graph by projecting the loop onto the base graph. Additional optimizations may take into account the degrees of nodes in the loops. In the case of matched lifted graphs (e.g., cyclically lifted graphs) a loop in the base graph is also a loop in the lifted Tanner graph precisely when the lifting values traversed in the loop reduce to the identity permutation.

[0139] According to certain aspects, using $j+3$ bit to represent the lifting and defining the lifting by taking the $j+3$ bit value modulo Z for Z in $2^j\{4,5,6,7\}$ results in a lifting for

$Z = 2^j 4$ given by the $j+2$ lower order bits and the higher order bit affects only the other 3 liftings.

[0140] For the optimization of the base graph for a set of clustered liftings, liftings values may be selected within a range $[0, (2^{j+4})-1]$. In other words, the lifting values may be selected from a range that is smaller than the smallest lifting size in the set of clustered liftings. Thus, in example described herein, for the tower of clustered liftings for $j=1$, the lifting size values may be selected from the range $[0:7]$.

[0141] For cyclically lifted graphs, each edge in the base graph has an associated integer as a lifting value. The value is taken positively when the edge is traversed in the variable-to-check direction and negatively in the check-to-variable direction. Given a loop in the base graph and a lifting size Z , the base loop will also be a lifted loop if the loop sum of the corresponding integers is 0 or has Z as a factor. Thus, when choosing integer value in the range $[0, 2^j 4]$ for the lifting values, the goal for $Z=2^j 4$ is to avoid summing to 0 or to having a factor of $2^j 4$ in the loop sum. For small loops, the sum generally will not be large, so in general, there are more such loops with a sum of magnitude $2^j 4$ than those with a sum of magnitude $2 \cdot 2^j 4$ or $3 \cdot 2^j 4$. Similarly, on average, sums of magnitude $2^j \{5, 6, 7\}$ and its multiples are less frequent. Thus, the small loop avoidance design problem is similar for these closely related values, where lift values in the range $[0:2^j 4]$ uses more than half the range available for $Z = 2^j \{5, 6, 7\}$. For a much larger Z , the used portion would be smaller and there may be a bigger gap between the best performance available for the large Z and that achievable by restricting liftings to a smaller Z . Thus, applying this approach over a relatively small range of Z values (e.g., within a factor of 2) is prudent. Hence, it is possible to find lift values that give good performance for four values simultaneously.

[0142] By utilizing a range of liftings which are numerically within a defined range along with an independent set of bits for each j with $j=1, \dots, 7$ the number of bits required is $3 + 4 + 5 + 6 + 7 + 8 + 9 = 42$ bits per edge to specify all of the liftings. By creating dependencies between different values of j this requirement may be further reduced. Additionally, often a structured LDPC graph will have special edges whose lifting values may be determined directly. For example, the edges connecting degree one variable nodes may always have lifting value 0. Edges on accumulate chains in encoding structures are also often set to 0. Such fixed lifting structure may not vary as the liftings vary and may

be referred to as having a special invariant structure. The lifting values for such edges can be more compactly represented. However, the number of edges having such a special invariant structure is a small portion of the total number of edges in the graph and does not significantly detract from the benefits of the above method for those edges that do not have a special invariant structure.

EXAMPLE: NESTED SCHEME FOR EFFICIENTLY LIFTING LDPC CODES

[0143] As described above, liftings in a clustered set of liftings (e.g., a “tower” of liftings”) can use the same lifting values (integers associated with the edge permutations) and, thus, the number of bits used to specify all of the liftings and lifting values may be reduced. This size reduction may allow for a reduced amount of memory for storing descriptions of all of the LDPC codes.

[0144] According to aspects of the present disclosure, a nested scheme for efficiently lifting LDPC codes may be used that further reduced the number of bits per edge in the base PCM.

[0145] As all liftings, even for different j values (e.g., liftings in different clustered sets), are based on the same base graph, the structures found to work for a small j value (i.e., for liftings in the corresponding set of clustered liftings) may be scaled and reused for larger j values (i.e., for larger liftings in another set). For example, a structure optimized for a smaller j may be retained and scaled for a larger j in order to reuse optimized bits found for the smaller j .

[0146] In one example, a transmitter/encoder device (e.g., such as a BS 110 or a UE 120) determines a base matrix that is associated with a cluster of lifting size values. The transmitting device selects a first lifting size value from the cluster of lifting size values for generating a lifted LDPC code by permutations of edges in the base matrix. The lifting size values in the cluster of lifting size values are within a defined range of each other. The transmitting device generates a first lifted matrix based on the base matrix and/or selected first lifting size value and selects a set of bits associated with the selected first lifting size value. The transmitting device selects a second lifting size value from the cluster of lifting size values and generates a second lifted matrix based on the base matrix, second selected lifting size value, and the set of bits. The transmitting device uses the generated second lifted matrix to generate the lifted LDPC code, encodes a

set of information bits based on the lifted LDPC code to produce a code word, and transmits the code word.

[0147] In the example described above, for $j=1$, the set of clustered liftings is $Z = \{8, 10, 12, 14\}$ may be designed using lifting values in the range $[0, 1, 2, \dots, 7]$. According to certain aspects, the liftings values selected for the $j=1$ graph can be multiplied by 2 and used for the $j=2$ graph, where the set of clustered liftings is $Z = \{16, 20, 24, 28\}$. In this case, the larger lifted graph (for $j=2$) inherits and improves on the loop structure of the smaller graph as the larger graph for lifting $2Z$ consists of two parallel copies of the original smaller graph with lifting Z . Because the smaller graph is designed to avoid loops summing to a factor of Z , it also avoids loops summing to factors of $2Z$. $j=1$ and $j=2$ are merely exemplary. In aspects, the lifting values for any set of clustered liftings may be used for another set of larger clustered liftings, and the lifting values can be multiplied by the factor of the difference in the liftings sizes of the two sets of liftings.

[0148] Further optimization of the larger graph could be achieved by altering the lowest order bit in the liftings. For example, after multiplication by 2 all liftings would have their lowest order bit set to 0. More generally, to achieve the best possible performance, more than just the lowest order bit may be altered. For example, two or three least significant bits may be altered. Generally, optimizing the three least significant bits results in nearly optimal performance. This preserves the large scale properties of the liftings (the most significant) bits, scaled up accordingly (by multiplying by 2) and then refines the details (the lower order bits) to find an optimal solution for the base graph for the next set of clustered liftings.

[0149] In one example, the three lowest order bits may be re-optimized. For the set of clustered liftings $j=1$, a 3-bit optimized lift per edge may be obtained. If the lifting values for an edge in the base graph (e.g., for the smallest lifting in the set $j=1$) are a, y , and z (i.e., 3 bits) in base 2 (i.e., where each of a, y , and z is an integer values of 0 or 1), then for the base graph for the set of clustered liftings $j=2$, the same edge will have lifting values of a, b, w, x , (i.e., 4 bits with one bit copied from the $j=1$ family) and in the base graph for the set of clustered liftings $j=3$, the edge will have lifting values a, b, c, u, v , (5 bits with 2 bits copied from the $j=2$ family) etc. Thus, the base graph for the set of clustered liftings $j=7$, the edge will have lifting value $a, b, c, d, e, f, g, r, s$ (i.e., 9 bits with 7 bits copied from the $j=6$ family) and the bits a, b, c, d, e, f, g are reused for smaller set of clustered

liftings j while the bits r and s are unique to $j=7$. The base graph for the set of clustered liftings uses j common bits and 2 unique bits. Thus, for all of the families $j=1 \dots 7$, there is a total of seven common bits and fourteen unique bits (i.e., 2 unique bits for each j), for a total of 21 bits to describe all seven code families. This is referred to as a “nested” scheme for describing the families of LDPC codes. If only the two lowest order bits were re-optimized then only 14 bits total would be needed. In some examples, most significant bits (MSBs) or any subset of consecutive bits can be used as common bits, rather than the LSBs. Both cases offer a substantial improvement on the 42-bit independent case.

[0150] As discussed above, certain structured LDPC graph may have a special invariant structure, for example, some special edges may have liftings that are invariant. For example, the 802.11 encoding structure, uses liftings of values 0 and 1. If this structure is retained, the structure is consistent with the above optimization of lower order bits only when at least two of the lower order bits are optimized. This is because $2 \times 1 = 2$; so if only the lowest order bit is optimized, the value 1 cannot be reached as only 2 and 3 are possible values. In this case, it may be preferable to retain the lifting value of 1. A similar technique can be used in which the low order bits are retained across different j and the higher order bits are re-optimized. In general, some bits from a smaller j may be reused to define values for the larger j while leaving enough bits for optimization so as to achieve good performance.

CONCLUSION

[0151] The encoding techniques described herein for high performance, flexible, and compact LDPC codes may lead to improved processor performance. For example, the techniques may allow for a processor to efficiently encode information of various blocklengths and code rates using good codes (e.g., having few loops). For example, a device, such as a processing system in BS 110 or UE 120 shown in FIG. 1, may encode and/or decode code words according to aspects of the present disclosure more quickly or more efficiently (e.g., consuming less power) than a device encoding and/or decoding code words according to previously known aspects.

[0152] The methods disclosed herein comprise one or more steps or actions for achieving the described method. The method steps and/or actions may be interchanged with one another without departing from the scope of the claims. In other words, unless a

specific order of steps or actions is specified, the order and/or use of specific steps and/or actions may be modified without departing from the scope of the claims.

[0153] As used herein, a phrase referring to “at least one of” a list of items refers to any combination of those items, including single members. As an example, “at least one of: a, b, or c” is intended to cover a, b, c, a-b, a-c, b-c, and a-b-c, as well as any combination with multiples of the same element (e.g., a-a, a-a-a, a-a-b, a-a-c, a-b-b, a-c-c, b-b, b-b-b, b-b-c, c-c, and c-c-c or any other ordering of a, b, and c).

[0154] As used herein, the term “determining” encompasses a wide variety of actions. For example, “determining” may include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, “determining” may include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, “determining” may include resolving, selecting, choosing, establishing and the like.

[0155] In some cases, rather than actually transmitting a frame, a device may have an interface to output a frame for transmission. For example, a processor may output a frame, via a bus interface, to an RF front end for transmission. Similarly, rather than actually receiving a frame, a device may have an interface to obtain a frame received from another device. For example, a processor may obtain (or receive) a frame, via a bus interface, from an RF front end for transmission.

[0156] The various operations of methods described above may be performed by any suitable means capable of performing the corresponding functions. The means may include various hardware and/or software component(s) and/or module(s), including, but not limited to a circuit, an application specific integrated circuit (ASIC), or processor. Generally, where there are operations illustrated in figures, those operations may have corresponding counterpart means-plus-function components with similar numbering.

[0157] For example, means for encoding, means for determining, means for selecting, and/or means for generating may include one or more processors, such as the TX MIMO processor 430, Transmit processor 420, and/or the Controller/Processor 440 of the BS 110 illustrated in FIG. 4; the TX MIMO processor 466, Transmit Processor 464, and/or the Controller/Processor 480 of the UE 120 illustrated in FIG. 4; and/or the encoder 1102 of the encoder 1100 illustrated in FIG. 11. Means for puncturing may comprise a processing

system, which may include one or more of processors of FIG. 4, and/or the puncturing module 1104 of the encoder 1100 illustrated in FIG. 11. Means for transmitting comprises a transmitter, which may include the Transmit processor 420, TX MIMO processor 430, modulator(s) 432a-432t, and/or the antenna(s) 434a-434t of the BS 110 illustrated in FIG. 4; the Transmit processor 464, TX MIMO Processor 466, modulator(s) 454a-454r, and/or antenna(s) 452a-452r of the UE 120 illustrated in FIG. 4; and/or the TX chain 1108 and antenna 1110 of the encoder 1100 illustrated in FIG. 11.

[0158] The various illustrative logical blocks, modules and circuits described in connection with the present disclosure may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device (PLD), discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any commercially available processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0159] If implemented in hardware, an example hardware configuration may comprise a processing system in a wireless node. The processing system may be implemented with a bus architecture. The bus may include any number of interconnecting buses and bridges depending on the specific application of the processing system and the overall design constraints. The bus may link together various circuits including a processor, machine-readable media, and a bus interface. The bus interface may be used to connect a network adapter, among other things, to the processing system via the bus. The network adapter may be used to implement the signal processing functions of the PHY layer. In the case of a wireless node (see FIG. 1), a user interface (e.g., keypad, display, mouse, joystick, etc.) may also be connected to the bus. The bus may also link various other circuits such as timing sources, peripherals, voltage regulators, power management circuits, and the like, which are well known in the art, and therefore, will not be described any further. The processor may be implemented with one or more general-purpose and/or special-purpose processors. Examples include microprocessors, microcontrollers, DSP processors, and

other circuitry that can execute software. Those skilled in the art will recognize how best to implement the described functionality for the processing system depending on the particular application and the overall design constraints imposed on the overall system.

[0160] If implemented in software, the functions may be stored or transmitted over as one or more instructions or code on a computer-readable medium. Software shall be construed broadly to mean instructions, data, or any combination thereof, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Computer-readable media include both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. The processor may be responsible for managing the bus and general processing, including the execution of software modules stored on the machine-readable storage media. A computer-readable storage medium may be coupled to a processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. By way of example, the machine-readable media may include a transmission line, a carrier wave modulated by data, and/or a computer readable storage medium with instructions stored thereon separate from the wireless node, all of which may be accessed by the processor through the bus interface. Alternatively, or in addition, the machine-readable media, or any portion thereof, may be integrated into the processor, such as the case may be with cache and/or general register files. Examples of machine-readable storage media may include, by way of example, RAM (Random Access Memory), flash memory, ROM (Read Only Memory), PROM (Programmable Read-Only Memory), EPROM (Erasable Programmable Read-Only Memory), EEPROM (Electrically Erasable Programmable Read-Only Memory), registers, magnetic disks, optical disks, hard drives, or any other suitable storage medium, or any combination thereof. The machine-readable media may be embodied in a computer-program product.

[0161] A software module may comprise a single instruction, or many instructions, and may be distributed over several different code segments, among different programs, and across multiple storage media. The computer-readable media may comprise a number of software modules. The software modules include instructions that, when executed by an apparatus such as a processor, cause the processing system to perform various functions. The software modules may include a transmission module and a receiving

module. Each software module may reside in a single storage device or be distributed across multiple storage devices. By way of example, a software module may be loaded into RAM from a hard drive when a triggering event occurs. During execution of the software module, the processor may load some of the instructions into cache to increase access speed. One or more cache lines may then be loaded into a general register file for execution by the processor. When referring to the functionality of a software module below, it will be understood that such functionality is implemented by the processor when executing instructions from that software module.

[0162] Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared (IR), radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, include compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk, and Blu-ray® disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Thus, in some aspects computer-readable media may comprise non-transitory computer-readable media (e.g., tangible media). In addition, for other aspects computer-readable media may comprise transitory computer-readable media (e.g., a signal). Combinations of the above should also be included within the scope of computer-readable media.

[0163] Thus, certain aspects may comprise a computer program product for performing the operations presented herein. For example, such a computer program product may comprise a computer-readable medium having instructions stored (and/or encoded) thereon, the instructions being executable by one or more processors to perform the operations described herein.

[0164] Further, it should be appreciated that modules and/or other appropriate means for performing the methods and techniques described herein can be downloaded and/or otherwise obtained by a wireless node and/or base station as applicable. For example, such a device can be coupled to a server to facilitate the transfer of means for performing the methods described herein. Alternatively, various methods described herein can be provided via storage means (e.g., RAM, ROM, a physical storage medium such as a

compact disc (CD) or floppy disk, etc.), such that a wireless node and/or base station can obtain the various methods upon coupling or providing the storage means to the device. Moreover, any other suitable technique for providing the methods and techniques described herein to a device can be utilized.

[0165] It is to be understood that the claims are not limited to the precise configuration and components illustrated above. Various modifications, changes and variations may be made in the arrangement, operation and details of the methods and apparatus described above without departing from the scope of the claims.

CLAIMS

1. A method for wireless communications, comprising:
 - selecting a first lifting size value Z and a first set of lifting values for generating a first lifted low density parity check (LDPC) code;
 - generating the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base parity check matrix (PCM) having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code;
 - determining a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values;
 - encoding a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and
 - transmitting the code word.
2. The method of claim 1, further comprising:
 - determining a tower of lifting size values by applying a range of powers of an integer to a range of lifting size values, wherein the tower of lifting size values comprises a plurality of sets of lifting size values.
3. The method of claim 2, wherein each set of lifting size values comprises the set of lifting size values corresponding to the range of powers of the integer applied to one lifting size value of the range of lifting size values.
4. The method of claim 2, wherein selecting the first lifting size value comprises selecting a largest lifting size value in each set of lifting size values.
5. The method of claim 2, further comprising:
 - storing only the lifted PCMs corresponding to the largest lifting size value in each set of lifting size values.
6. The method of claim 2, wherein:
 - the second lifting size value is in a same set as the first lifting size value; and

determining the second set of lifting values is based on a function involving the first set of lifting values and the second lifting size value.

7. The method of claim 6, wherein determining the second set of lifting values comprises:

for each lifting value in the second set of lifting values, performing a modulo operation, wherein the second lifting size value is a divisor and a corresponding lifting value in the first set of lifting values is a dividend.

8. The method of claim 6, wherein determining the second set of lifting values comprises performing a floor operation based on the second lifting size value and the first set of lifting values.

9. The method of claim 1, wherein encoding the set of information bits comprises: selecting a lifted LDPC code based on a target range of blocklengths of the code word.

10. The method of claim 1, wherein:
the first lifted PCM and the second lifted PCM are associated with a first code rate;
and
the method further comprises generating different lifted PCMs for at least one of the first lifting size value or the second lifting size value based on second base PCM associated with a second rate.

11. The method of claim 10, wherein the second base PCM is generated by adding hybrid automatic repeat request (HARQ) extension bits to the first base PCM.

12. The method of claim 2, wherein:
the range of lifting size values comprises lifting size values within a defined range of each other.

13. The method of claim 12, wherein the defined range of lifting size values comprises lifting size values within a factor of each other.

14. The method of claim 12, wherein each set of lifting size values in the tower of lifting size values corresponds to a power of the range of powers of the integer applied to each of the range of lifting size values.
15. The method of claim 14, wherein selecting the first lifting size value comprises selecting a smallest lifting size value in one of the sets of lifting size values.
16. The method of claim 12, further comprising:
using the base PCM for the first lifted PCM for the other lifting size values in that set of lifting size values.
17. The method of claim 14, further comprising:
determining lifting values for other sets of lifting size values in the tower of lifting size values based on the first set of lifting values.
18. The method of claim 17, wherein determining the lifting values for an edge in the lifted PCMs for the different set of lifting size values comprises:
copying a portion of bits of the lifting value in the first set of lifting values associated with that edge; and
adding a number of unique bits to the portion of copied bits.
19. The method of claim 18, wherein the first portion of copied bits is equal to the number of the sets in the tower and the number of unique bits for each set is equal to the integer.
20. The method of claim 14, further comprising:
determining a second base PCM and a third set of lifting size values for another set of lifting size values in the tower of lifting size values; and
using the second base PCM and the third set of lifting values for other lifting size values in that set of lifting size values.
21. An apparatus for wireless communications, comprising:
means for selecting a first lifting size value Z and a first set of lifting values for generating a first lifted low density parity check (LDPC) code;

means for generating the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base parity check matrix (PCM) having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code;

means for determining a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values;

means for encoding a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and

means for transmitting the code word.

22. The apparatus of claim 21, further comprising:

means for determining a tower of lifting size values by applying a range of powers of an integer to a range of lifting size values, wherein the tower of lifting size values comprises a plurality of sets of lifting size values, wherein each set of lifting size values comprises the set of lifting size values corresponding to the range of powers of the integer applied to one lifting size value of the range of lifting size values.

23. The apparatus of claim 22, wherein:

the second lifting size value is in a same set as the first lifting size value; and
determining the second set of lifting values is based on a function involving the first set of lifting values and the second lifting size value.

24. The apparatus of claim 22, wherein:

the range of lifting size values comprises lifting size values within a defined range of each other.

25. The apparatus of claim 22, further comprising:

means for using the base PCM for the first lifted PCM for the other lifting size values in that set of lifting size values.

26. The apparatus of claim 22, further comprising:

means for determining lifting values for other sets of lifting size values in the tower of lifting size values based on the first set of lifting values.

27. The apparatus of claim 26, wherein means for determining the lifting values for an edge in the lifted PCMs for the different set of lifting size values comprises:

means for copying a portion of bits of the lifting value in the first set of lifting values associated with that edge; and

means for adding a number of unique bits to the portion of copied bits.

28. An apparatus for wireless communications, comprising:

at least one processor coupled with a memory and configured to:

select a first lifting size value Z and a first set of lifting values for generating a first lifted low density parity check (LDPC) code;

generate the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base parity check matrix (PCM) having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code;

determine a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values; and

encode a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and a transmitter configured to transmit the code word.

29. The apparatus of claim 28, wherein the at least one processor is further configured to:

determine a tower of lifting size values by applying a range of powers of an integer to a range of lifting size values, wherein the tower of lifting size values comprises a plurality of sets of lifting size values, wherein each set of lifting size values comprises the set of lifting size values corresponding to the range of powers of the integer applied to one lifting size value of the range of lifting size values.

30. A computer readable medium having computer executable code stored thereon for wireless communications, comprising:

code for selecting a first lifting size value Z and a first set of lifting values for generating a first lifted low density parity check (LDPC) code;

code for generating the first lifted LDPC code by applying the first set of lifting values to interconnect edges in Z copies of a base parity check matrix (PCM) having a first number of base variable nodes and a second number of base check nodes to obtain a first lifted PCM corresponding to the first lifted LDPC code;

code for determining a second set of lifting values for generating a second lifted PCM corresponding to a second lifted LDPC code for a second lifting size value based on the first lifted PCM and the first set of lifting values;

code for encoding a set of information bits based on at least one of: the first lifted LDPC code or the second lifted LDPC code to produce a code word; and

code for transmitting the code word.