

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
30 April 2009 (30.04.2009)

PCT

(10) International Publication Number
WO 2009/053846 A2

(51) International Patent Classification:

H04L 1/00 (2006.01)

(21) International Application Number:

PCT/IB2008/003452

(22) International Filing Date: 30 July 2008 (30.07.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/952,736	30 July 2007 (30.07.2007)	US
60/957,056	21 August 2007 (21.08.2007)	US
60/957,061	21 August 2007 (21.08.2007)	US
60/978,218	21 August 2007 (21.08.2007)	US

(71) Applicant (for all designated States except US): **MARVELL ISRAEL (MISL) LTD.** [IL/IL]; 6 Hamada Street, Mordot HaCarmel Industrial Park, 20692 Yokneam (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SPENCER, Paul, S.** [IL/IL]; 26 Yosef Street, Modiin (IL). **AMIR, Winstok** [IL/IL]; Derech Hashalom 81B, Flat 6, Tel Aviv (IL).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

(54) Title: RATE MATCHING FOR A WIRELESS COMMUNICATIONS SYSTEM

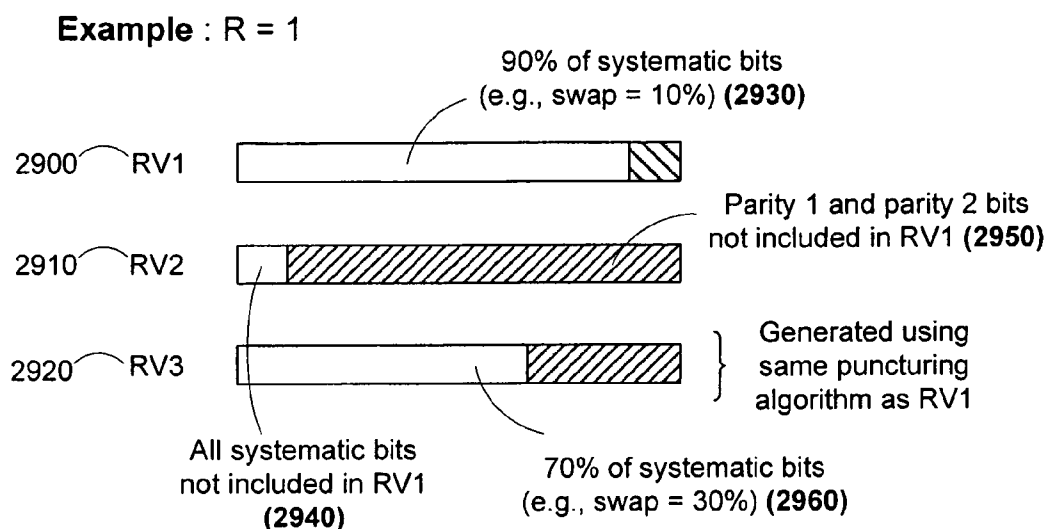


FIG. 29

(57) **Abstract:** Apparatuses and methods are provided for generating a plurality of redundancy versions using various rate matching algorithms. In some embodiments, a rate matcher is provided that allocates systematic and parity bits to the redundancy versions in a manner that allows all of these bits to be transmitted in at least one redundancy version. In some embodiments, the rate matcher uses a first puncturing algorithm to generate both a first redundancy version and a third redundancy version, but allocates a different proportion of the systematic bits to these redundancy versions. In these embodiments, the second redundancy version may include only bits that were not transmitted in the first redundancy version.

RATE MATCHING FOR A WIRELESS COMMUNICATIONS SYSTEM

Cross-Reference to Related Applications

[0001] This claims the benefit under 35 U.S.C. §119(e) of U.S. Provisional Application Nos. 60/952,736 filed
5 July 30, 2007, 60/957,056, filed August 21, 2007,
60/957,061, filed August 21, 2007, and 60/978,218, filed
August 21, 2007, which are hereby incorporated by
reference herein in their entirety.

Background of the Disclosure

10 [0002] The disclosed technology relates generally to
transmitting and receiving messages over a wireless
communications network, and more particularly to
distributing encoded data across a plurality of
redundancy versions for transmission and receipt over the
15 wireless communications network.

[0003] In accordance with various wireless
communications standards, each base station and mobile
station employs a rate matcher. The rate matcher
determines which of the bits in an encoded message to
20 send in a current transmission block. For reasons

- 2 -

related to data integrity, and because the encoded message may be substantially larger than the amount of data that can be transmitted in a single transmission block, a transmitting station (e.g., a base station or a mobile station) may perform multiple transmissions based on the same encoded message before the receiving station (e.g., a mobile station or a base station) can recover the message. The rate matcher, therefore, determines which set of bits to select for each transmission block.

10 **[0004]** There are several known rate matching algorithms that are employed by cellular communications systems today. These algorithms have several weaknesses. One weakness is the high percentage of bits in the encoded message that are not selected for transmission within a reasonable number of transmissions. Another weakness is the drastic decrease in decoding performance if the receiving station is not able to interpret the header in one of the transmissions. Accordingly, it would be desirable to provide a rate matcher that is

20 capable of a high decoding performance in different operating scenarios.

Summary of the Disclosure

[0005] Accordingly, apparatuses and methods for rate matching are disclosed. A rate matcher configured in accordance with the present invention may be implemented within a transmitting station and/or a receiving station in a wireless communications system, or in any other suitable communications or storage system. For purposes of clarity, and not of limitation, the rate matcher may be described below as being implemented on a base station

- 3 -

as part of a transmitting station and as being implemented on a mobile station as part of a receiving station. However it should be understood that both the transmitting stations and the receiving stations of the disclosed embodiment can be implemented on both base stations and mobile stations.

[0006] A transmitting station constructed in accordance with the present invention may include a Turbo encoder, a rate matcher, and a transmitter. The Turbo encoder may employ a rate-1/3 mother code, and can encode the message to produce encoded data. The encoded data may include a plurality of systematic bits and a plurality of parity bits (e.g., parity 1 and parity 2 bits). The rate matcher may then generate a plurality of redundancy versions of the message by selecting bits from the systematic, parity 1, and parity 2 bits for inclusion in each redundancy version according to a rate matching algorithm. The transmitter may then transmit the plurality of the redundancy versions over the wireless communications network using, for example, a hybrid automatic repeat request (HARQ) protocol.

[0007] A receiving station constructed in accordance with the present invention may include a receiver, a rate matcher, and a Turbo decoder. The receiver may receive a plurality of the redundancy versions that were transmitted from a transmitting station over the wireless communications network. The rate matcher may then process the received redundancy versions to determine which of the bits of the redundancy versions correspond to systematic, parity 1, and parity 2 bits of an encoded

- 4 -

message. The rate matcher of the receiving station may process the received redundancy versions using the same rate matching algorithm used by the rate matcher of the transmitting station. The Turbo decoder may then decode
5 the encoded message to produce an estimate of the encoded data.

[0008] In some embodiments of the present invention, the rate matcher may select a first subset of the systematic bits using a first puncturing algorithm and a
10 first parameterized swap value. The size of the first subset may be based at least partly on the first parameterized swap value. The rate matcher may then generate a first redundancy version, having a predetermined number of bit positions. The first
15 redundancy may include a first subset of the systematic bits. The unused bit positions of the first redundancy version may be allocated to a first subset of the parity bits.

[0009] The rate matcher may then select a second
20 subset of the systematic bits. The second subset may include all of the systematic bits corresponding to the message that were not included in the first subset of the systematic bits. The rate matcher may generate a second redundancy version of the message, which has the
25 predetermined number of bit positions, may include the second subset of the systematic bits. The unused bit positions of the second redundancy version may be allocated to a second subset of parity bits that includes parity bits what are not included in the first subset of
30 parity bits.

- 5 -

- [0010] In some embodiments, a rate matcher and/or transmitter and/or receiver may operate based on a particular transmission scheme selected from among a plurality of available transmission scheme. For example, a transmitting station may include a configuration unit that is configured to select the particular transmission scheme and to provide the corresponding system-level rate matching parameters to the rate matcher and/or the corresponding transmission parameters to the transmitter. The rate matcher, which may operate based on the selected transmission scheme, generates a plurality of redundancy versions, where each of the plurality of systematic bits is allocated to at least one of the redundancy versions. The systematic bits may also be allocated such that each redundancy version includes less than all of the systematic bits. The transmitter, which may operate based on the selected transmission scheme, may transmit the plurality of redundancy versions over the wireless communications network to a receiver which may include a rate matcher which may operate on the selected transmission scheme. The transmitter transmits at least one of the redundancy versions over a plurality of time slots defined by a time division multiple access (TDMA) protocol.
- [0011] In still other embodiments of the present invention, a rate matcher may operate such that all systematic bits are included in the first two redundancy versions of a message. In particular, the rate matcher may select a first subset of the systematic bits and may generate a first decodable redundancy version of the

- 6 -

message that includes the first subset of systematic bits. The rate matcher may then select a second subset of the systematic bits that includes all of the systematic bits corresponding to the message that are not
5 included in the first subset of the systematic bits. The rate matcher may then generate at least one additional redundancy version of the message including the second subset of systematic bits.

[0012] In some embodiments of the present invention, a
10 rate matcher may generate redundancy versions such that any parity bits selected for the first redundancy version is excluded from the second subset of the parity bits. In particular, the rate matcher may select a first subset of the parity bits. The rate matcher may generate a
15 first redundancy version including the first subset of parity bits. The rate matcher may then select a second subset of the parity bits such that those parity bits selected for the first subset are excluded from being selected for the second subset. In some embodiments,
20 after all of the parity bits that were not selected for the first subset have been selected for the second subset, the parity bits that were selected for the first subset may also be selected for the second subset. Using the second subset of parity bits, the rate matcher may
25 generate a second redundancy version.

[0013] In still other embodiments, a rate matcher can generate redundancy versions such that pairs of redundancy versions are decodable. In particular, the rate matcher may select a first subset of the systematic
30 bits. The rate matcher may generate a first decodable

- 7 -

redundancy version of the message that includes the first subset of systematic bits. The rate matcher may then select a second and a third subset of the systematic bits. The rate matcher may allocate all of the

5 systematic bits corresponding to the message that are not included in the first subset to at least one of the second and third subsets. The rate matcher may then generate a second redundancy version of the message that includes the second subset and a third redundancy version

10 of the message that includes the third subset. From the allocation of systematic bits to the second and third redundancy versions, a combination of the second and third redundancy versions may be decodable without any additional systematic information provided from the first

15 redundancy version.

[0014] In other embodiments of the present invention, a rate matcher configured in accordance with the present invention can generate at least three redundancy versions of an encoded message, where the third redundancy version

20 is the same as the first redundancy version. First, the rate matcher may select a first subset of the systematic bits. The rate matcher may generate a first decodable redundancy version of the message that includes the first subset of systematic bits. The rate matcher may then

25 select a second subset of the systematic bits, where the second subset includes all of the systematic bits corresponding to the message that were not included in the first redundancy version. The rate matcher may generate a second redundancy version of the message that

30 includes the second subset of systematic bits. The rate

- 8 -

matcher may then generate a third redundancy version of the message that includes the first subset of systematic bits.

[0015] In still other embodiments of the present invention, a rate matcher may generate at least three redundancy versions, where two of the redundancy versions are generated using the same puncturing algorithm, but different swap values. In particular, the rate matcher may select a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value. The rate matcher may generate a first redundancy version that includes the first subset of systematic bits. Then, the rate matcher may select a second subset of the systematic bits using a second puncturing algorithm. The second subset can include all of the systematic bits that were not included in the first redundancy version. The rate matcher may generate a second redundancy version, where the second redundancy version includes the second subset of the systematic bits. The rate matcher may then select a third subset of the systematic bits using the first puncturing algorithm and a second parameterized swap value. The second parameterized swap value is different from (e.g., greater than) the first parameterized swap value such that the third subset includes a different number of systematic bits than the first redundancy version. Using the third subset of systematic bits, the rate matcher may generate a third redundancy version

[0016] In still other embodiments of the present invention, a rate matcher configured in accordance with

- 9 -

the present invention may implement loop logic for puncturing data bits. The rate matcher may initialize a first set of loop parameters for puncturing the systematic bits. Using the first set of loop parameters as input, the rate matcher may execute the loop logic to puncture a subset of the systematic bits. The rate matcher may also initialize a second set of loop parameters for puncturing the first plurality of parity bits. Using the second set of loop parameters as input, the rate matcher may execute the loop logic to puncture a subset of the first plurality of parity bits. Finally, the rate matcher may combine the systematic bits and the first plurality of parity bits that have not been punctured.

Brief Description of the Drawings

[0017] The above and other aspects and advantages of embodiments of the invention will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which like reference characters refer to like parts throughout, and in which:

[0018] FIG. 1 is a simplified schematic diagram of a radio cell in a cellular network that includes a base station and a mobile station, each having an enhanced rate matcher employing the disclosed technology;

[0019] FIG. 2 is a simplified block diagram of a base station with control circuitry that employs the enhanced rate matching techniques of the disclosed technology;

[0020] FIG. 3 is a simplified block diagram of a mobile station with control circuitry that employs the

- 10 -

enhanced rate matching techniques of the disclosed technology;

[0021] FIG. 4 is a simplified block diagram of the communications circuitry for a base station, which
5 includes an enhanced rate matcher that employs the disclosed technology;

[0022] FIG. 5 is a simplified block diagram of a communications circuitry for a mobile station, which includes an enhanced rate matcher that employs the
10 disclosed technology;

[0023] FIG. 6 is a flowchart of an illustrative process for transmitting encoded data in two redundancy versions using a hybrid automatic repeat request (ARQ) protocol, where all of the systematic bits are included
15 in the first redundancy version (RV1), in accordance with an embodiment of the present invention;

[0024] FIG. 7 is a flowchart of an illustrative process for transmitting encoded data in three redundancy versions using a HARQ protocol, where all of the
20 systematic bits are included in the first redundancy version (RV1), in accordance with an embodiment of the present invention;

[0025] FIG. 8 is a schematic diagram illustrating two redundancy versions generated in accordance with the
25 flowchart of FIG. 6;

[0026] FIG. 9 is a schematic diagram illustrating three redundancy versions generated in accordance with the flowchart of FIG. 7;

- 11 -

[0027] FIG. 10 is a flowchart of an illustrative process for selecting systematic and parity bits to include in the last redundancy version;

5 [0028] FIG. 11 is a flowchart of an illustrative process for decoding a plurality of received redundancy versions that collectively include all bits of an encoded message, where the plurality of redundancy versions are received using a HARQ protocol;

10 [0029] FIG. 12A is a flowchart of an illustrative process for using a *flip* variable when two redundancy versions are need to transmit all bits of an encoded message;

15 [0030] FIG. 12B is a flowchart of an illustrative process for using a *flip* variable when three redundancy versions are needed to transmit all bits of an encoded message, where the parity bits that were included in a first redundancy version are excluded from the second redundancy version using the *flip* variable;

20 [0031] FIG. 13 shows two tables of rate matching parameters, where the first table illustrates *flip* variables for different redundancy versions and bit types (systematic versus parity) and the second table illustrates various transmission schemes that use non-zero *swap* values;

25 [0032] FIG. 14 is a flowchart of an illustrative process for generating two redundancy versions, where a non-zero *swap* value is used such that neither redundancy version includes all of the systematic bits;

30 [0033] FIG. 15 is a flowchart of an illustrative process for generating three redundancy versions, where a

- 12 -

non-zero *swap* value is used such that none of the redundancy versions includes all of the systematic bits;

[0034] FIG. 16 is a schematic diagram illustrating two redundancy versions generated in accordance with the flowchart of FIG. 14;

[0035] FIG. 17 is a schematic diagram illustrating three redundancy versions generated in accordance with the flowchart of FIG. 15;

[0036] FIG. 18 is a flowchart of an illustrative process for generating a redundancy version by separately initializing loop parameters and separately running loop logic for the systematic, parity 1, and parity 2 bits;

[0037] FIG. 19 is a table of loop parameters computed based on internal parameters and bit type (systematic, parity 1, or parity 2);

[0038] FIG. 20 is a flowchart of illustrative loop logic for looping through and selectively puncturing systematic, parity 1, and parity 2 bits in accordance with an embodiment of the present invention;

[0039] FIG. 21 is a flowchart illustrating the result of performing the loop logic of FIG. 20 when only two redundancy versions are needed (e.g., $R_{\max} = 2$);

[0040] FIG. 22 is a flowchart illustrating the result of performing the loop logic of FIG. 20 when three redundancy versions are needed (e.g., $R_{\max} = 3$);

[0041] FIG. 23 is a flowchart showing the loop logic in FIG. 20 with additional steps for puncturing fewer bits when a piggybacked ACK/NACK (PAN) field is not included in a radio block;

- 13 -

[0042] FIG. 24 is a flowchart of an illustrative process for distributing systematic bits between three redundancy versions (RV1, RV2, and RV3) such that RV1 is decodable and RV2 and RV3 are decodable together;

5 [0043] FIG. 25 is a schematic diagram illustrating three redundancy versions generated in accordance with the flowchart of FIG. 24;

[0044] FIG. 26 is a flowchart of an illustrative process for generating three redundancy versions such that the third redundancy version is the same as the first redundancy version;

[0045] FIG. 27 is a schematic diagram illustrating three redundancy versions generated in accordance with the flowchart of FIG. 26;

15 [0046] FIG. 28 is a flowchart of an illustrative process for generating three redundancy versions such that RV1 and RV3 are generated using the same puncturing algorithm but different swap values;

[0047] FIG. 29 is a schematic diagram illustrating three redundancy versions generated in accordance with the flowchart of FIG. 28;

[0048] FIG. 30 is a table of *flip* variables for different redundancy versions for use with the flowchart of FIG. 26;

25 [0049] FIG. 31 is a flowchart of illustrative loop logic for looping through and selectively puncturing systematic, parity 1, and parity 2 bits in accordance with an embodiment of the present invention;

- 14 -

[0050] FIG. 32 is a flowchart illustrating the result of performing the loop logic of FIG. 31 when three redundancy versions are needed (e.g., $R_{\max} = 3$); and

[0051] FIGS. 33A and 33B are a flowchart of an
5 illustrative process for decoding one or more received redundancy versions at a receiving station.

Detailed Description of the Disclosure

[0052] FIG. 1 shows an illustrative wireless communications system 100, which may be a cellular
10 system. Wireless communications system 100 can include a plurality of base stations that are interconnected to form a mobile or cellular network. For example, wireless communications system 100 can include base station 102. Base station 102 may be configured to communicate with
15 mobile stations within its radio communication range via radio signals. Radio cell 101 may represent a physical area that lies within the radio communication range of base station 102.

[0053] Base station 102 may be able to communicate
20 with mobile station 106, which is located within radio cell 101, and with other mobile stations that are also within radio cell 101 (not shown). For example, base station 102 may broadcast information that can be received by any mobile station within radio cell 101,
25 including mobile station 106. Such transmissions from base station 102 to mobile station 106 (or any other mobile station) may sometimes be referred to as downlink transmissions.

[0054] Mobile station 106 may be any suitable type of
30 cellular telephone compatible with the mobile network of

- 15 -

wireless communications system 100. For example, mobile station 106 can operate based on a protocol or communications standard compatible with base station 102. In some embodiments, the mobile network may use the

5 EGPRS2 Downlink ("DL") protocol. The EGPRS2 DL protocol may define, for example, various coding and modulation properties that allow cellular communication to occur reliably and at a high data rate. In other embodiments, the mobile network illustrated in FIG. 1 can employ a

10 protocol other than a EGPRS2 DL protocol, such as a modified form of the EGPRS2 DL protocol. For example, the mobile network may employ a modified protocol that may sometimes be referred to as the Reduced symbol Duration, Higher Order modulation and Turbo codes (RED

15 HOT) protocol.

[0055] Base station 102 may send an encoded message via downlink transmissions to mobile station 106 in blocks of a predetermined size. The predetermined size may be defined by the transmission protocol employed by

20 the cellular network. To select which parts of the encoded message to transmit in each block, base station 102 may include enhanced rate matcher 104. As will be described in greater detail below, enhanced rate matcher 104 may select portions of the encoded message

25 using particular selection techniques to generate one or more redundancy versions of the message. These selection techniques include puncturing algorithms that puncture the encoded data such that the encoded data is distributed between multiple redundancy version of the

30 message. These redundancy versions of the message may

- 16 -

then be transmitted to mobile station 106. Mobile station 106 may include enhanced rate matcher 108 to interpret the one or more redundancy versions of the message received from base station 102. In particular, enhanced rate matcher 108 may apply the same selection technique used by rate matcher 104 to determine which portion (e.g., which bit locations) of the encoded message were included in each of the received redundancy versions of the message. In some embodiments, mobile station 106 may send encoded messages via uplink transmissions to base station 102 using a similar technique.

[0056] FIGS. 2 and 3 show illustrative block diagrams of a base station and a mobile station, respectively, constructed in accordance with an embodiment of the present invention. Referring first to FIG. 2, a block diagram of base station 200 is shown which may be a more detailed representation of base station 102 of FIG. 1. Base station 200 can include signal processing and/or control circuits 252, mass data storage 264, memory 266, antenna 251 for wireless transmission and/or reception, and network interface 220. Mass data storage 264 and memory 266 may be implemented using any suitable type of storage mediums (e.g., ROM, RAM, FLASH, etc.), and may be coupled to signal processing and/or control circuits 252 to support the functions of signal processing and/or control circuits 252.

[0057] Base station 200 may obtain a message from a base station controller that controls the flow of information over the cellular network. In particular,

- 17 -

base station 200 may include network interface 220 that communicates with the base station controller over communications link 222. From the base station controller, network interface 220 may acquire the message
5 to transmit to a particular mobile station (e.g., mobile station 106 of FIG. 1). Network interface 220 may provide the message to signal processing and/or control circuits 252 for processing and transmission to the mobile station.

10 **[0058]** To transmit the message, signal processing and/or control circuits 252 may include enhanced rate matcher 253. Enhanced rate matcher 253 may have any of the features or functionalities of rate matcher 104 (FIG. 1). For example, enhanced rate matcher 253 may select
15 portions of the encoded message to generate one or more redundancy versions of the message. As will be described in greater detail below, enhanced rate matcher 253 may select portions of the encoded message using puncturing algorithms that puncture the encoded data such that the
20 encoded data is distributed between multiple redundancy version of the message. Signal processing and/or control circuits 252 may further include any communication circuitry (not shown) needed to transmit data selected by enhanced rate matcher 253 via antenna 251. In some
25 embodiments, signal processing and/or control circuits 252 may also perform coding and/or encryption, perform calculations, format data and/or perform other base station functions. Signal processing and/or control circuits 252 may include a single processor, or may
30 include several processors each of which is configured to

- 18 -

perform one or more base station functions. For example, signal processing and/or control circuits 252 may include a modem processor and an application processor. The processors may be hardware, software, or firmware-based
5 processors.

[0059] Referring now to FIG. 3, a block diagram of mobile station 300 is shown which may be a more detailed representation of mobile station 106 of FIG. 1. Mobile station 300 can include mass data storage 364,
10 memory 366, WLAN interface 368, signal processing and/or control circuits 352, antenna 351 for wireless transmission and/or reception, microphone 356, audio output 358, display 360, and user input device 362. Generally, each of these components may be controlled,
15 receive information, or provide information to signal processing and/or control circuits 352.

[0060] The signal processing and/or control circuits 352 and/or other circuits (not shown) in the cellular telephone 350 may include any suitable
20 communications circuitry (e.g., a modem) (not shown) coupled to antenna 351 to receive and demodulate incoming data from a base station (e.g., base station 102 of FIG. 1 or base station 200 of FIG. 2). For example, antenna 351 may receive signals corresponding to the message
25 transmitted from antenna 251 of base station 200 (FIG. 2). To interpret the data received via antenna 351, signal processing and/or control circuits 352 may include enhanced rate matcher 353. Enhanced rate matcher 353 may have any of the features or functionalities of enhanced
30 rate matcher 108 of FIG. 1. For example, signal

- 19 -

processing and/or control circuits 352 may also process data, perform coding and/or encryption, perform calculations, format data and/or perform other cellular telephone functions. Signal processing and/or control

5 circuits 352 may include a single processor, or may include several processors each of which is configured to perform one or more cellular telephone functions. For example, signal processing and/or control circuits 32 may include a modem processor and an application processor.

10 The processors may be hardware, software, or firmware-based processors. Although base station 102 of FIG. 1 and base station 200 of FIG. 2 have been described above as encoding and transmitting messages to mobile stations and mobile station 106 of FIG. 1 and mobile station 300

15 of FIG. 3 have been described above as receiving and decoding messages from base stations, it should be understood that that both base stations and mobile stations may transmit, receive, and process messages from base stations and/or mobile stations.

20 **[0061]** Turning to FIGS. 4 and 5, simplified block diagrams of signal processing and/or control circuits are shown for transmitting and receiving data, respectively, in a base station and/or a mobile station. Referring first to FIG. 4, a block diagram of signal processing

25 and/or control circuits 400 a transmitting station is shown. Signal processing and/or control circuits 400 may be a more detailed representation of signal processing and/or control circuits 252 of base station 200 (FIG. 2) or signal processing and/or control

30 circuits 352 of mobile station 300 (FIG. 3). Signal

- 20 -

processing and/or control circuits 400 may include Turbo encoder 410, rate matcher 420, interleaver 430, partitioner 440, and transmitter 450. These components may be configured to convert message 405 to radio signal
5 455 that is suitable for a wireless transmission.

[0062] Turbo encoder 410 may encode message 405 to produce encoded message 415. Turbo encoder 410 may operate using any suitable Turbo code, such as a rate-1/3 Turbo mother code that converts message 405, having N_{sys}
10 bits, to an encoded message that has approximately $3N_{sys} + 12$ bits also referred to herein as $3N_w$ bits, where N_w is equal to $N_{sys} + 4$. Turbo encoder 410 may be a systematic encoder that outputs the bits of the original message 405, referred to as a stream of "systematic
15 bits," as well as two streams of N_{sys} parity bits. For simplicity, one stream of N_{sys} parity bits may be referred to as "parity 1" bits, and the other stream of N_{sys} parity bits may be referred to as "parity 2" bits. The remaining bits produced by Turbo encoder 410 (e.g., the
20 remaining 12 bits) may be termination bits.

[0063] Enhanced rate matcher 420 may select a portion of the systematic, parity 1, and parity 2 bits for transmission to a receiving station (e.g., mobile station 106 of FIG. 1 or mobile station 300 of FIG. 3).
25 Enhanced rate matcher 420 may have any of the features or functionalities of enhanced rate matcher 104 (FIG. 1) or enhanced rate matcher 253 (FIG. 2) or enhanced rate matcher 353 (FIG. 3). Thus, for each downlink transmission, enhanced rate matcher 420 may select a
30 predetermined number of these bits for transmission based

- 21 -

on an employed transmission scheme or on transmission constraints and limitations. The number of bits selected by enhanced rate matcher 420 may be referred to by the variable, M , which can be of any suitable size. Thus, enhanced rate matcher 420 may use a suitable selection technique for selecting M bits as data 425 for transmission from approximately $3N_w$ bits.

[0064] Since only a portion of encoded message 415 is selected for each transmission, multiple transmissions may be used in order to send most or all of the encoded message 415 and in order to achieve successful reception in less than optimal conditions. Enhanced rate matcher 420 may select a different set of M bits for each transmission, where the sets may be overlapping or non-overlapping sets. Thus, enhanced rate matcher 420 may produce different M -bit versions of encoded message 415. These versions may be referred to as "redundancy versions" (RVs) of a message or encoded message. Alternatively, the redundancy versions may be referred to as "puncturing versions," since the selection of M bits may be performed by puncturing bits from the approximately $3N_w$ bits. The first redundancy version may be referred to as RV1, and the second redundancy version may be referred to as RV2, etc.

[0065] The M bits 425 of encoded message 405 selected for the next transmission may then be passed to interleaver 430. Interleaver 430 may interleave the M bits selected for transmission such that neighboring bits are separated from one another. In some embodiments, the M bits may be transmitted in multiple blocks, sometimes

- 22 -

referred to "bursts." In these embodiments, interleaver 430 may interleave the M bits such that neighboring bits are moved to adjacent or non-adjacent bursts. Interleaver 430 therefore produces interleaved data 435, which is the same as data 425 but arranged in a different order. Partitioner 440 may then partition the interleaved data 435 into distinct blocks for transmission. For example, partitioner 440 may partition interleaved data 435 into four transmission blocks 445.

10 **[0066]** Transmitter 450 prepares each of transmission blocks 445 for transmission over the wireless network. Transmitter 450 may include communications circuitry (not shown), such as a suitable modulator, to convert blocks 445 to convert the bits of transmission blocks 445

15 into radio signal 455 suitable for wireless transmission. The modulator may use any suitable modulation scheme to perform the conversion (e.g., quadrature amplitude modulation (QAM), pulse amplitude modulation (PAM), phase shift keying (PSK), etc.), which may be defined by the

20 transmission scheme. Transmitter 450 may employ a time division multiple access (TDMA) protocol in which each block of blocks 445 is allocated to a different time slot. Thus, each redundancy version may be transmitted in radio signal 455 in four time slots.

25 **[0067]** In some embodiments, transmitter 450 may append a header to each of the transmission blocks 445 prior to transmission. The header may provide information on which messages are being sent in the current time slot - in this case, message 405. Each message in transmission

30 block 445 may be referred to sometimes as a block

- 23 -

sequence number (BSN). This allows a receiving station receiving the radio signal to identify which is being sent, and to match the redundancy version that is received to any redundancy versions for the same message
5 that have already been received.

[0068] FIG. 4 illustrates the signal processing and/or control circuit that transmits one BSN in each radio block. In some embodiments of the present invention, the signal processing and/or control circuit may be
10 configured to transmit redundancy versions for a plurality of encoded messages (e.g., two, three, or four encoded messages, etc.) in the same radio block.

[0069] Referring now to FIG. 5, a block diagram of signal processing and/or control circuits 500 for a
15 receiving station is shown. Signal processing and/or control circuits 500 may be a more detailed representation of signal processing and/or control circuits 252 of base station 200 (FIG. 2) or signal processing and/or control circuits 352 of mobile
20 station 300 (FIG. 3). Signal processing and/or control circuits 500 may include receiver 510, departitioner 520, deinterleaver 530, enhanced rate matcher 540, and Turbo decoder 555. The operation of these components may complement the operation of transmitter 450, partitioner
25 440, interleaver 430, enhanced rate matcher 420, and Turbo encoder 410 of FIG. 4, respectively, where each component of signal processing and/or control circuits 500 can essentially undo the processing of its corresponding component in signal processing and/or

- 24 -

control circuits 400 to reconstruct a message that was transmitted using enhanced rate matching techniques.

[0070] For example, receiver 510 may receive radio signal 505 from a downlink wireless transmission that corresponds to a burst of information (e.g., part of a redundancy version). Receiver 510 may include a demodulator/detector that interprets radio signal 505. Thus, for each time slot, receiver 510 may produce a stream of bits corresponding to all or part of a redundancy version. Once all parts of a redundancy version have been received (e.g., all four bursts), departitioner 520 may combine the streams to produce data 525. Data 525 includes all the bits of the redundancy version, and therefore has M bits. Deinterleaver 530 may then deinterleave data 525 to obtain the original order of the sequence.

[0071] From deinterleaved data 535, rate matcher 540 may identify which M bits of the original $3N_w$ bits in the encoded message have been received. In particular, enhanced rate matcher 540 may determine how the received M bits of a received redundancy version were selected from the $3N_w$ bits in the encoded message in order to produce an estimate 545 of the encoded message. In addition, if previous redundancy versions have been received for the same message, estimate 545 of the encoded message may add information from the previously received redundancy versions to generate an improved estimate of the encoded message. Turbo decoder 550 may then decode estimate 545 of the encoded message using a code corresponding to the the same Turbo code employed by

- 25 -

the corresponding Turbo encoder (e.g., Turbo encoder 410 of FIG. 4). The resulting output of Turbo decoder 550 is an estimate 555 of the original message. If the transmission is successful, estimate 555 is the same as the original message (e.g., message 405 of FIG. 4).

5 **[0072]** The success of Turbo decoder 550 depends at least partially on whether estimate 545 includes information on some, most, or all of the bits from the encoded message. In particular, although Turbo decoder 10 550 may be able to successfully decode an encoded message estimate 545 with only partial information, it may do so with a higher rate of success if estimate 545 includes information on a greater number of the bits in the encoded message, rather than having to use erasures for 15 some of the bit positions. Therefore, in some embodiments, redundancy versions may be generated such that each bit of the encoded message is selected for at least one of the redundancy versions. Therefore, in the event that all of the redundancy versions are 20 transmitted, each bit estimate in the encoded message estimate 545 may be based on information from at least one of the redundancy versions.

25 **[0073]** FIG. 6 shows an illustrative flowchart of process 600 for transmitting all of the bits of an encoded message in two redundancy versions using a hybrid automatic repeat request (HARQ) protocol. All of the bits in the encoded message can be transmitted in two redundancy versions when:

$$2M \geq 3N_w, \quad (\text{EQ. 1})$$

- 26 -

so that all $3N_w$ bits in the systematic, parity 1, and parity 2 streams can be transmitted within the $2M$ bits of two redundancy versions. EQ. 1 may be rewritten as:

$$N_w/M \leq 2/3, \quad (\text{EQ. 2})$$

5 where this ratio may be referred to as the code rate and sometimes by the variable, R . Therefore, process 600 may be executed in embodiments where the code rate is less than or equal to 0.66. The rate matcher may also initialize a variable, $R_{\max}$, that represents the number of
10 redundancy versions needed to transmit the full encoded message. Thus, when $R \leq 0.66$, the rate matcher may set $R_{\max}$ to two.

[0074] Process 600 may begin at step 602 and continue to step 604. In an embodiment, at step 604, all of the
15 systematic bits and a portion of the parity bits (e.g., parity 1 and/or parity 2 bits) may be selected for transmission in a first redundancy version of a message over the wireless communications network. In this embodiment, all of the systematic bits may be included in
20 the first redundancy version so that the corresponding receiving station can attempt to recover the message using all of the systematic bits, should the first redundancy version be properly received and its header successfully decoded. The receiving station's Turbo
25 decoder may be more likely to correctly recover the original message when the Turbo decoder has information on all of the systematic bits. Parity bits may be selected for the first redundancy version such that a total of M bits are chosen for transmission. This
30 selection of parity bits may be performed by a rate

- 27 -

matcher implemented on the transmitting station. This process for selecting systematic and parity bits is merely illustrative of an embodiment, other techniques for selecting bits for inclusion in one or more
5 redundancy versions will be described in greater detail below.

[0075] Process 600 may then continue to step 606. At step 606, it may be determined whether an acknowledgment (ACK) signal has been received from the receiving
10 station. If an ACK is received, the receiving station has successfully recovered the transmitted message, and therefore process 600 may move to step 608 and end. Otherwise, if a NACK signal is received or if no ACK signal is received within a predetermined period of time,
15 the message may not have been successfully decoded by the receiving station. Thus, process 600 may continue to step 606 in which a second redundancy version is generated for transmission.

[0076] In the second redundancy version of the
20 message, the transmitting station may avoid selecting parity bits that were included in the first redundancy version, until all of the parity bits have been selected for transmission in at least one redundancy version. Then, other bits that were included in the first
25 redundancy version at step 604 may be selected for inclusion in the second redundancy version. Thus, after the second redundancy version is transmitted at step 606, the transmitting station has transmitted all of the bits in the encoded message at least once.

- 28 -

[0077] Process 600 may then proceed to step 612. At step 612, the transmitting station may determine whether an ACK signal has been received following transmission of the second redundancy version. If the transmitting station receives an ACK signal, the receiving station has successfully recovered the message. Thus, process 600 may proceed to step 608 and end. Otherwise, process 600 may return to step 604, where the transmitting station may retransmit the first redundancy version, for example.

10 The transmitting station may retransmit the first and the second redundancy versions in this manner until an ACK signal is received from the receiving station or until the transmitting station retransmits the first and second redundancy versions a predetermined number of times.

15 [0078] Turning now to FIG. 7, an illustrative flowchart of process 700 is shown for including for transmission over a wireless communications network all of the bits of an encoded message in three redundancy versions using a HARQ retransmission protocol. Suitable HARQ transmission protocols are described in the 3rd

20 Generation Partnership Project (3GPP) technical specifications (TS), for example, at section 43.064, release 7, version 7.8.0. This portion of the 3GPP TS is incorporated by reference herein in its entirety. This process for selecting systematic and parity bits for is

25 merely illustrative of an embodiment, other techniques for selecting bits for inclusion in one or more redundancy versions will be described in greater detail below. A suitable transmitting station may execute the

30 steps of process 700 when $R_{\max} = 3$ and when:

- 29 -

$$3M \geq 3N_w > 2M, \quad (\text{EQ. 3})$$

so that all $3N_w$ bits in the systematic, parity 1, and parity 2 streams can be transmitted within the $3M$ bits of three redundancy versions, but cannot be transmitted in two redundancy versions. EQ. 3 may be rewritten as:

$$0.66 < N_w/M \leq 1 \quad (\text{EQ. 4})$$

Therefore, the transmitting station may execute the steps of process 700 when the code rate of the system is greater than 0.66, but less than or equal to one.

10 **[0079]** Process 700 begins at step 702, and then proceeds to step 704. At step 704, the transmitting station may select all of the systematic bits and a portion of the parity bits (e.g., parity 1 and/or parity 2 bits) for transmission in a first redundancy version of a message over the wireless communications network. As
15 described above in connection with step 604 of process 600 (FIG. 6), the transmitting station may include all of the systematic bits in the first redundancy version to maximize the probability that a corresponding receiving
20 station can recover the transmitted message in a single transmission.

25 **[0080]** Then, at step 706, the transmitting station may determine whether an ACK signal has been received from the receiving station. If an ACK signal has been received, the receiving station has successfully recovered the message, and process 700 may end at step 708. Otherwise, if a NACK signal is received or if no ACK signal is received within a predetermined period of time, the message may not have been successfully decoded
30 by the receiving station. Therefore, process 700 may

- 30 -

continue to step 708, where the transmitting station may generate a second redundancy version for transmission.

[0081] For the second redundancy version of the message, the transmitting station may select, at
5 step 712, a portion of the parity bits that were not included in the first redundancy version. In an embodiment, parity bits that were selected for transmission in the first redundancy version are excluded from selection for the second redundancy version, unless
10 all of the previously unselected bits are included in the second redundancy version. Thus, if the receiving station successfully receives both the first and the second redundancy versions, the receiving station can perform decoding based on all of the systematic bits
15 (from the first redundancy version) and the parity bits included in either the first or the second redundancy versions. In some embodiments, the transmitting station may also include some of the systematic bits that were included in the first redundancy version in subsequent
20 redundancy versions. Therefore, for some of the systematic bits of the encoded message, the receiving station can perform decoding using information from multiple transmissions.

[0082] Process 700 may then continue to step 712 in
25 which the transmitting station determines whether an ACK signal has been received following transmission of the second redundancy version. If the transmitting station determines that an ACK signal has been received, process 700 may move to step 708 and end. Otherwise, the
30 receiving station still may not have successfully

- 31 -

recovered the message, and therefore process 700 moves to step 714.

[0083] At step 714, the transmitting station transmits a third redundancy version of the message. In the third
5 redundancy version, the transmitting station may generate for transmission a redundancy version that includes all the parity bits that were not included in the first or the second redundancy versions. Thus, following receipt of the third redundancy version the receiving station may
10 have information on all of the systematic, parity 1, and parity 2 bits. Since the transmitting station may include a total of M bits for transmission in the third redundancy version, in some embodiments, the transmitting station may allocate any remaining bit positions to some
15 of the bits (e.g., systematic, parity 1, and/or parity 2 bits) that were included in the first redundancy version.

[0084] Then, process 700 may move to step 718, where the transmitting station may again determine whether an ACK signal has been received. If an ACK is received,
20 process 700 may end at step 708. If a NACK signal or no acknowledgment is received, process 700 may return to step 704 and retransmit the first redundancy version. Thus, the transmitting station may transmit the first, second, and third redundancy versions in order until an
25 ACK signal is received from the receiving station, or until the transmitting station has retransmitted these redundancy versions a predetermined number of times.

[0085] It should be noted that the flowcharts of processes 600 (FIG. 6) and 700 (FIG. 7) are merely
30 illustrative. Any of the steps may be modified, removed,

- 32 -

or combined, or any additional steps may be modified, without departing from the scope of the present disclosure. For example, the steps of either flowchart may be modified to use a retransmission protocol other
5 than a HARQ protocol for transmitting the different redundancy versions.

[0086] Referring now to FIGS. 8 and 9, two examples are shown illustrating the redundancy versions of a message that may be generated from executing
10 processes 600 (FIG. 6) and 700 (FIG. 7). In particular, FIG. 8 illustrates the rate matching that generates two redundancy versions when the code rate of the system is 0.50, and FIG. 9 illustrates the rate matching that generates three redundancy versions when the code rate of
15 the system is 0.75. It should be understood that the examples of FIGS. 8 and 9, as well as any other similar rate matching illustrations in later figures, merely illustrate the constitution of redundancy versions. The illustrations are not intended to illustrate the order of
20 the bits in each redundancy version, and should not be limited to such.

[0087] FIG. 8 illustrates the generation of redundancy versions 830 and 840 from systematic bits 800, parity 1 bits 810, and parity 2 bits 820. The code rate in this
25 example is 0.5, and therefore, systematic bits 800, parity 1 bits 810, and parity 2 bits 820 are each half the bit length of M , the number of bits in each redundancy version. After rate matching, a first redundancy version, RV1 830, is produced, where half of
30 the bit positions include all systematic bits 850, and

- 33 -

the remaining half are allocated to parity bits 860 in some suitable proportion. Systematic bits 850 may be the same as systematic bits 800, while parity bits 860 may be allocated between parity 1 bits 810 and parity 2 bits 820
5 in some suitable proportion (e.g., half parity 1 bits and parity 2 bits).

[0088] The second redundancy version, RV2 840, may exclude all parity bits 840 that were included in RV1 830 until there are no more unallocated parity bits.
10 Therefore, due to the 0.5 code rate, half of the bit positions in RV2 840 are allocated to previously unused parity bits. The remaining bit positions 880 in RV2 are used for bits that were included in RV1 830. For example, remaining bit positions 880 may include all of
15 systematic bits 850, all of parity bits 860, or part of both systematic bits 850 and parity bits 860 in some suitable proportion. One way for a rate matcher to select bits for RV2 840 will be described below in connection with FIG. 10.

20 [0089] FIG. 9 illustrates the generation of redundancy versions 930, 940, and 950 from systematic bits 900, parity 1 bits 910, and parity 2 bits 920. The code rate in this example is 0.75, and therefore, systematic bits 900, parity 1 bits 910, and parity 2 bits 920 are
25 each $3/4$ of bit length of M , the number of bits in each redundancy version. After rate matching, a first redundancy version, RV1 930, is produced, where $3/4$ of the bit positions therein are allocated to systematic bits 960, and the remaining quarter of the bit positions
30 in RV1 930 are allocated to parity bits 970 in some

- 34 -

suitable proportion. Systematic bits 960 may be the same as systematic bits 900, while the parity bits may be allocated to parity 1 bits 910 and parity 2 bits 920 in some suitable proportion (for example, half parity 1
5 bits, and half parity 2 bits).

[0090] The second redundancy version, RV2 940, may include parity bits 970 which are selected from parity 1 bits 910 and/or parity 2 bits 920. In an embodiment, parity bits 970 may initially exclude parity bits that
10 were included in RV1 930. In some embodiments, in addition to parity bits 970, RV2 may include a portion of the systematic bits that were included in RV1, although this may not be the case.

[0091] Turning to the third redundancy version in
15 FIG. 9, RV3 950 includes parity bits 980, which is composed of parity 1 bits 910 and/or parity 2 bits 920 that were not included in RV1 930 or RV2 940. For a code rate of 0.75, there may still be space in RV3 950 even after all of the previously unused parity bits are
20 selected for parity bits 980. In an embodiment, these remaining bit positions 990 are allocated to a portion of the bits that were included in RV1 930. For example, remaining bit positions 990 may include only systematic bits 900, only parity 1 bits 910 and/or parity 2 bits
25 920, or both systematic bits and parity bits in some suitable proportion.

[0092] FIG. 10 shows an illustrative flowchart of process 1000 for selecting bits for inclusion in a final redundancy version (e.g., RV2 for systems with $R \leq 0.66$,
30 RV3 for systems with $R > 0.66$). The steps of process 1000

- 35 -

may be a more detailed view of step 610 in process 600 (FIG. 6) or step 714 in process 700 (FIG. 7). The steps of process 1000 may be performed by a suitable rate matcher (e.g., enhanced rate matcher 104 of FIG. 1, etc.), and may be the steps used by the rate matcher to generate RV2 840 in FIG. 8 or RV3 950 of FIG. 9.

[0093] Process 1000 begins at step 1002 and may continue to step 1004. At step 1004, the rate matcher may select parity bits (e.g., parity 1 bits and/or parity 2 bits) that were not included in a first redundancy version. Thus, after the final redundancy version is transmitted, the corresponding receiving station may have information on all of the bits in an encoded message. After these parity bits are selected, there may still be remaining bit positions available in the final redundancy version. Therefore, at step 1006, the rate matcher may choose at least a subset of the systematic bits that were included in the first redundancy version. For example, the rate matcher may choose all of the systematic bits for inclusion in the final redundancy version, or may choose only a portion of the systematic bits that were included in a previous redundancy version. Process 1000 may then continue to step 1008 in which the rate matcher partitions any remaining bit positions in the final redundancy version to parity 1 bits and parity 2 bits that were included in the first redundancy version. For example, the rate matcher may allocate approximately half of the remaining bits to parity 1 bits, and may allocate the remaining

- 36 -

approximately half to parity 2 bits. Process 1000 then moves to step 1010 and ends.

[0094] As described above, in one embodiment, when a rate matcher generates three redundancy versions, any remaining space in the third redundancy version, after all of the parity bits that were not included in the other redundancy version are added to RV3, is allocated to bits included in RV1. In one embodiment, 30% of the systematic bits of the encoded message are included in RV3. In other embodiments of the present invention, the rate matcher may allocate at least some of the remaining space to bits included in RV2. The selection of bits for inclusion in RV3 may be done so as to maximize the probability that the corresponding receiving station can recover the transmitted message even if the first redundancy version is not successfully received (e.g., due to failure to interpret the header).

[0095] Referring now to FIG. 11, an illustrative flowchart of process 1100 is shown for decoding received redundancy versions at a receiving station (e.g., mobile station 106 of FIG. 1 or mobile station 300 of FIG. 3) using a HARQ protocol. The redundancy versions received by the receiving station may have been generated using any of the techniques described above in connection with FIGS. 4-10.

[0096] Process 1100 begins at step 1102 and proceeds to step 1104. At step 1104, the receiving station decodes the header portion of a received packet. The header may include a BSN which identifies the message being transmitted in the data portion of the received

- 37 -

packet. At step 1106, the receiving station determines whether the decoding is successful. For example, the receiving station may determine whether the BSN has been successfully identified from decoding the header. If the
5 decode is not successful, process 1100 ends at 1108 without the receiving station obtaining any information from the transmitted redundancy version.

[0097] If, at step 1106, the receiving station determines that the header portion has been successfully
10 decoded, process 1100 moves to step 1107. At step 1107, the receiving stations may identify the bits of the received redundancy version. In particular, a rate matcher within the receiving station may be used to identify which bit positions of the encoded message are
15 included in the received redundancy version. The rate matcher in the receiving station may identify these bits using the same bit selection technique applied by corresponding rate matcher in the transmitting station to generate the redundancy versions. (Step 1107 is
20 described in more detail with respect to the process illustrated below in FIG. 34) Then, at step 1110, the receiving station may determine whether other redundancy versions associated with the same message have been received. For example, the receiving station may compare
25 the BSN of the received redundancy version with the BSN of previously received redundancy versions. If other redundancy versions have not been received, process 1100 skips step 1112 and proceeds to step 1114. At step 1114, the receiving station decodes the data portion of the
30 received packet. That is, when step 1112 is skipped, the

- 38 -

receiving station performs decoding using only one redundancy version.

[0098] If, at step 1110, the receiving station instead determines that other redundancy versions have been
5 received, process 1100 continues to step 1112 in which the receiving station combines the data portions (e.g., the redundancy versions) of all the received packets. For example, the receiving station may use incremental redundancy (IR) combining to combine bits that are
10 included in one redundancy version but not other redundancy versions, and may use Chase combining or any other symbol-level or bit-level combining to combine bits that are included in multiple redundancy versions. If the receiving station has received and successfully
15 decodes the headers of all two (for $R \leq 0.66$) or all three (for $0.66 < R \leq 1$) redundancy versions, the receiving station has information on all of the bits in the encoded message. Thus, at step 1114, the receiving station can decode the combined redundancy version that
20 has information on the full encoded message.

[0099] After decoding at step 1114, process 1116 continues to step 1116. At step 1116, the receiving station determines whether decoding the data portion was successful. For example, the receiving station may
25 determine whether any errors can be detected in the decoded message. If the receiving station determines that decoding was not successful, the receiving station may perform any steps necessary after decoding failure at step 1112. For example, the receiving station may
30 discard the erroneously decoded packet, or the receiving

- 39 -

station may save the erroneously decoded packet to combine with the next redundancy version that is received. Then, at step 1124, the receiving station may transmit a NACK back to the transmitting station to
5 request transmission of another redundancy version. The receiving station may transmit the NACK signal with other packet data that the receiving station needs to transmit to the transmitting station. Therefore, the NACK signal may be transmitted in a "piggy-backed ACK/NACK" (PAN)
10 field of the packet data. Process 1100 may then return to step 1104 to decode the next packet received from the transmitting station.

[0100] Returning to step 1116, if the receiving station determines instead that the data portion of the
15 received packet has been successfully decoded, process 1100 may move to step 1118. At step 1118, the receiving station may pass on the decoded data portion from the communications circuitry of the receiving station to modules of the receiving station that use this
20 information. The receiving station may also transmit, at step 1120, an ACK signal back to the transmitting station to inform the transmitting station that the message has been successfully received. In some embodiments, the receiving station may transmit the ACK signal in a PAN
25 field. Process 1100 may then end at step 1108.

[0101] In some embodiments, a variable, referred to as a *flip* variable, may be used by a rate matcher within a transmitting station or receiving station when generating each redundancy version. The value of the *flip* variable
30 may cause the rate matcher to exclude or include

- 40 -

particular bits, typically parity bits, in a redundancy version. For example, the rate matcher may set *flip* to zero to exclude bits from being selected for a current redundancy version that have already been included in a previous redundancy version. The rate matcher may instead set *flip* to one so that all bits that were not included in a previous redundancy version may be selected in the current redundancy version. Using the *flip* variable, the rate matcher may generate redundancy versions where all bits in the encoded message are selected for inclusion in at least one redundancy version, such as the redundancy versions illustrated in FIGS. 8 and 9.

[0102] FIG. 12A shows an illustrative flowchart of process 1200 in which a rate matcher within a suitable transmitting or receiving station uses a *flip* variable to select parity bits for inclusion in two redundancy versions (e.g., when $R \leq 0.66$). The rate matcher may select systematic bits in the manner described above, where all of the systematic bits are chosen for inclusion in the first redundancy version, or may select systematic bits using any of the techniques described below. Thus, while process 1200 is directed to the selection of parity bits, it should be understood that systematic bits are also included in the redundancy versions, and can be selected using any of a variety of approaches. In some embodiments, the selection of parity bits proceeds independently of the selection of systematic bits.

[0103] Process 1200 may begin at step 1202 and continue to step 1204. At step 1204, a rate matcher, may

- 41 -

select a first subset of the parity bits for inclusion in a first redundancy version. The parity bits may include parity 1 bits and parity 2 bits in any suitable proportion (e.g., half parity 1 bits and half parity 2 bits). Then, at step 1206, the rate matcher may generate the first redundancy version that includes the first subset of the parity bits, as well as all or a portion of the systematic bits. The rate matcher may keep track of which bits were included in RV1 at step 1210. For example, the rate matcher may maintain a binary vector that indicates which parity bits were included in RV1.

[0104] Process 1200 may then continue to step 1212, where the rate matcher may select a second subset of parity bits for inclusion in a second redundancy versions. When only two redundancy versions are generated, the rate matcher may choose all of the parity bits (e.g., parity 1 and/or parity 2 bits) that were not included in the first redundancy version for inclusion in the second redundancy version. For example, the rate matcher may initialize the *flip* variable to one, and may use the binary vector generated at step 1210 to ensure that all of the bits not selected for RV1 are now selected for RV2.

[0105] At step 1214, the rate matcher, may generate the second redundancy version. The second redundancy version includes the second subset of parity bits selected at step 1212, as well as all or a portion of the systematic bits. Therefore, after the second redundancy version is generated at step 1214, all of the parity bits

- 42 -

may be included in at least one of the two redundancy versions. Process 1200 may then end at step 1218.

[0106] FIG. 12B shows an illustrative flowchart of process 1250 in which a rate matcher within a suitable transmitting or receiving station uses a *flip* variable to select parity bits for inclusion in three redundancy versions (e.g., when $0.66 < R \leq 1$). The rate matcher may select systematic bits in the manner described above, where all of the systematic bits are included in the first redundancy version, or may select systematic bits using any of the techniques described below. Thus, while process 1250 is directed to the selection of parity bits, it should be understood that the systematic bits can be included in the redundancy versions and selected using any of a variety of approaches.

[0107] Process 1250 begins at step 1252 and proceeds to step 1254. At step 1254, the rate matcher, may select a first subset of the parity bits for inclusion in a first redundancy version. The first subset of parity bits may include both parity 1 bits and parity 2 bits. Then, at step 1256, the rate matcher may generate the first redundancy version that includes the first subset of the parity bits, as well as all or a portion of the systematic bits. The rate matcher may keep track of which bits were included in RV1 at step 1260. For example, the rate matcher may maintain a binary vector that indicates which parity bits were included in RV1.

[0108] Process 1250 may then continue to step 1262, where the rate matcher may select a second subset of parity bits for inclusion in a second redundancy version.

- 43 -

The rate matcher may select only those parity bits for the second subset that were not included in the first redundancy version. Note, however, that if additional parity bits are required to complete a redundancy version, and all of the parity bits are included either in the first subset or the second subset, then parity bits included in the first subset may also be selected again for inclusion in the second subset. In particular, step 1262 may include step 1263, where the rate matcher excludes those bits that were included in RV1 from being selected for the second subset. To execute step 1263, the rate matcher may initialize the *flip* variable to zero, which may cause the rate matcher to automatically discard any of the parity bits that the binary vector generated at step 1260 indicates has already been included in RV1. By including only those parity bits that have not already been included in another redundancy version, the rate matcher ensures that all of the remaining parity bits not selected for RV1 or RV2 will fit into the *M* bits of RV3.

[0109] Then, at step 1264, the rate matcher may generate the second redundancy version. The second redundancy version includes the second subset of parity bits selected at step 1262, as well as all or a portion of the systematic bits. At step 1260, the rate matcher may keep track of which parity bits have now been included in either RV1 or RV2. For example, the rate matcher may generate a binary vector that indicates which parity bits have already been included in a redundancy version.

- 44 -

[0110] Process 1250 may then continue to step 1268, where the rate matcher generates a third redundancy version. The rate matcher may generate the third redundancy version using a similar technique as that used to generate the second redundancy version when $R \leq 0.66$, as described above in connection with FIG. 12A. In particular, the rate matcher may select a third subset of the parity bits that includes all of the parity bits (e.g., parity 1 and/or parity 2 bits) that were not included in RV1 or RV2. For example, the transmitting station's rate matcher may initialize the *flip* variable to one, and may use the binary vector generated at step 1260 to ensure that all of the bits not selected for RV1 or RV2 are now selected for RV3. In accordance with an embodiment, if all of the parity bits are already included in one of RV1 or RV2, then the technique used to select parity bits for RV1 may be repeated, or some other suitable technique employed. Thus, after the third redundancy version is generated at step 1267, all of the parity bits may be included in at least one of the three redundancy versions. Process 1150 may then end at step 1272.

[0111] It should be noted that process 1200 (FIG. 12A) and 1250 (FIG. 12B), as well as any other processes provided in this disclosure, are merely illustrative. Any of the steps of these processes may be modified, combined, or removed, or any additional steps may be added, without departing from the scope of the present invention. For example, steps may be added to implement

- 45 -

a HARQ protocol, or any other retransmission protocol, for transmitting the different redundancy versions.

[0112] FIG. 13 shows Table 1 of values for the *flip* variable that are initialized by a rate matcher when
5 generating redundancy versions. Table 1 includes a row for each of the different data streams - e.g., the streams of systematic bits, parity 1 bits, and parity 2 bits. The columns of Table 1 represent four different types of redundancy versions that may be generated by the
10 rate matcher. In particular, when $R_{\max} = 2$, the rate matcher may generate a first redundancy version (RV1) and a type-I second redundancy version (RV2-I). When $R_{\max} = 3$, the rate matcher may generate a first redundancy version (RV1), a type-II second redundancy version (RV2-II), and
15 a third redundancy version (RV3). For the parity 1 and parity 2 bits, the values in Table 1 correspond to those described above in connection with process 1200 (FIG. 12A) and 1250 (FIG. 12B). For example, in accordance with an embodiment, *flip* may be initialized to
20 one for RV2-I so that all parity bits not previously included in RV1 are selected for RV2-II. On the other hand, the *flip* variable may be set to zero for RV2-II so that all bits that were included in RV1 are not selected for RV2-II.

25 [0113] Table 1 shows that the *flip* variable may be initialized to different values for systematic bits than for parity bits in some embodiments of the present invention. The particular *flip* values shown in Table 1 for the systematic bits enable the rate matcher to
30 generate redundancy versions similar to those shown in

- 46 -

FIGS. 8 and 9, described above. In particular, by initializing the *flip* variable to one in RV2-I, RV2-II, and RV3, the rate matcher does not exclude systematic bits that were included in the first redundancy version from being selected for any of the later redundancy versions.

[0114] In some embodiments, a rate matcher selects only a portion of the systematic bits to include in a first redundancy version. In these embodiments, the rate matcher may allocate the remaining systematic bits to the second redundancy version, or to both the second redundancy version and the third redundancy version (if $R_{\max} = 3$). The rate matcher may select a sufficiently large proportion of the systematic bits (e.g., 95%, 90%, 85%, etc.) for inclusion in the first redundancy version to maintain a high probability that a receiving station will be able to successfully decode the first redundancy version. By leaving some of the systematic bits for the second and/or third redundancy versions, the probability that a receiving station can successfully recover the message after receiving the second or third redundancy version is increased, particularly in the event that the header portion of the first redundancy version is not successfully decoded by the receiving station.

[0115] The rate matcher may define a variable, referred to as a *swap* variable, that represents the percentage of the systematic bits that are not included in the first redundancy version, but rather in one or more of the other redundancy versions. In other words, the swap value corresponds to a partitioning of

- 47 -

systematic bits, between a first redundancy version and another redundancy version. FIG. 13 shows Table 2 of rate matching parameters for a plurality of transmission schemes (e.g., DAS-5, DAS-10, DBS-5, DBS-6, DBS-8, where
5 DAS-5 and DAS-10 are EGPRS2 Downlink level A modulation and coding schemes and DBS-5, DBS-6, and DBS-8 are EGPRS2 Downlink level B modulation and coding schemes) available for use by the transmitting station and corresponding receiving station. These transmission schemes may
10 represent some or all of the available transmission schemes that have non-zero swap values. In some embodiments, the transmitting station and receiving station may also have available transmission schemes with zero-valued *swap* variables. The first column of Table 2
15 includes the *swap* values for each of the transmission schemes having non-zero swap values. In this table, some of the transmission schemes have a *swap* value of 5%, indicating that 5% of the systematic bits are removed, or "punctured," from the first redundancy version and
20 included in one of the other redundancy versions. The remaining transmission schemes have a *swap* value of 15%, indicating that 15% of the systematic bits are punctured from the first redundancy version.

[0116] The available transmission protocols may also
25 define the particular Turbo coding and modulation scheme used by the transmitting station and receiving station, as well as other transmission parameters. Thus, the remaining columns of Table 2 show the number of systematic bits used in the transmission protocol, as
30 well as the number of total bits that can be included in

- 48 -

each redundancy version when a PAN field is not included ($M_{\text{NO_PAN}}$) and when a PAN field is included (M_{PAN}).

[0117] FIGS. 14 and 15 show illustrative flowcharts of processes for generating redundancy versions using non-zero swap values when $R_{\text{max}} = 2$ and $R_{\text{max}} = 3$, respectively. Referring first to FIG. 14, the flowchart for process 1400 is shown, which may be executed by any suitable rate matcher (e.g., enhanced rate matcher 253 of FIG. 2 or enhanced rate matcher 353 of FIG. 3). Process 1400 begins at step 1402 and proceeds to step 1404. At step 1404, the rate matcher may initialize a *swap* value. The rate matcher may initialize the *swap* value based on the particular transmission protocol being used. The *swap* value may be a zero-valued swap value or any suitable non-zero-valued swap value (e.g., 5%, 15%, 30%, etc.). For example, if the rate matcher employs a DAS-5 transmission protocol with the rate matching parameters shown in the first row of Table 2 (FIG. 13), the rate matcher may initialize the *swap* variable to 5%.

[0118] Then, at step 1406, the rate matcher may select a first subset of the systematic bits using a first puncturing algorithm and the swap value. The percentage of the systematic bits selected for the first subset is based on the value of the *swap* value initialized at step 1402. For example, if the rate matcher employs the DAS-5 transmission protocol, the rate matcher may select $466 * 0.95 = 443$ systematic bits for inclusion in the first subset. At step 1408, the rate matcher may generate a first redundancy version that includes the first subset of systematic bits. The remaining bits

- 49 -

available in the first redundancy version are allocated to parity bits (e.g., parity 1 and/or parity 2 bits). For example, if the rate matcher transmitting station employs the DAS-5 protocol without PAN, the $M_{NO_PAN} = 1248$ bits of RV1 may include the 443 systematic bits selected at step 1406 and 805 parity bits.

[0119] Process 1400 may then continue to step 1412, where the rate matcher selects a second subset of the systematic bits using a second puncturing algorithm. The second subset of systematic bits includes all the remaining systematic bits that were not included in the first redundancy version. For example, in accordance with an embodiment, if the transmitting station employs the DAS-5 protocol, the rate matcher may include the 23 bits that were not selected for RV1 in the second subset. Then, at step 1414, the rate matcher may generate a second redundancy version that includes the second subset of systematic bits and all parity bits based on a flip parameter as described above. For example, in an embodiment, the second redundancy version may be generated so that it includes all of the parity bits not included in RV1. Any space remaining in the second redundancy version may be allocated to the systematic and/or parity bits that were included in the first redundancy version. Process 1400 may then end at step 1418.

[0120] FIG. 15 shows an illustrative flowchart of process 1500 for generating redundancy versions using a non-zero swap value when $R_{max} = 3$. The steps of process 1500 may be executed by any suitable rate matcher

- 50 -

(e.g., enhanced rate matcher 253 of FIG. 2 or enhanced rate matcher 353 of FIG. 3). Process 1500 begins at step 1502 and proceeds to step 1504. At step 1504, the rate matcher initializes a *swap* value. The rate matcher
5 may initialize the *swap* value based on the particular transmission protocol being used, and the *swap* value may be a zero-valued swap value or any suitable non-zero-valued swap value (e.g., 5%, 15%, 30%, etc.). For example, in accordance with an embodiment, if a rate
10 matcher employs a DBS-6 transmission protocol with a PAN field, and with the rate matching parameters shown in the fourth row of Table 2 (FIG. 13), the rate matcher may initialize the *swap* variable to 15%.

[0121] Then, at step 1506, in accordance with an
15 embodiment, the rate matcher may select a first subset of the systematic bits using a first puncturing algorithm and the swap value. The percentage of the systematic bits selected for the first subset is based on the value of the *swap* value initialized at step 1402. For example,
20 if a rate matcher employs the DBS-6 transmission protocol, the rate matcher may select $610 * 0.85 = 519$ systematic bits for inclusion in the first subset. At step 1508, the rate matcher may generate a first redundancy version that includes the first subset of
25 systematic bits. The remaining bits available in the first redundancy version are allocated to parity bits (e.g., parity 1 and/or parity 2 bits). For example, if a transmitting station employs the DAS-5 protocol with PAN, the $M_{PAN} = 876$ bits of RV1 may include the 519 systematic
30 bits selected at step 1506 and 357 parity bits.

- 51 -

[0122] Process 1500 may then continue to step 1512, where the rate matcher selects a second subset of the systematic bits using a second puncturing algorithm. The second subset of systematic bits includes all the
5 remaining systematic bits that were not included in the first redundancy version. For example, if a rate matcher employs the DBS-6 protocol, the rate matcher may include the 91 systematic bits that were not selected for RV1 in the second subset. Then, at step 1514, the rate matcher
10 may generate a second redundancy version that includes the second subset of systematic bits, as well as a portion of the parity bits (e.g., parity 1 and/or parity 2 bits) that were not included in the first redundancy version.

15 **[0123]** At step 1518, the rate matcher may generate a third redundancy version using a third puncturing algorithm. In an embodiment, the third redundancy version may include all of the parity bits that were not included in the first redundancy version or the second
20 redundancy version. In some embodiments, any remaining bit locations in the third redundancy version, after all of the systematic, parity 1, and/or parity 2 bits have been included in one of the redundancy versions, may be allocated to any of the bits (e.g., systematic, parity 1, and/or parity 2 bits) that were included in RV1.
25 Process 1500 may end at step 1522.

[0124] Referring now to FIGS. 16 and 17, two examples are shown illustrating the redundancy versions that may be generated by executing processes 1400 (FIG. 14)
30 and 1500 (FIG. 15). In particular, FIG. 16 illustrates

- 52 -

the rate matching that generates two redundancy versions when the code rate of the system is 0.5 and the *swap* value is 25%, and FIG. 17 illustrates the rate matching that generates three redundancy versions when the code rate of the system is 1.0 and *swap* value is 25%.

5 **[0125]** FIG. 16 illustrates the generation of redundancy versions 1630 and 1640 from systematic bits 1600, parity 1 bits 1610, and parity 2 bits 1620. 75% of systematic bits 1600 are selected for inclusion in
10 RV1 1630. The remaining bit positions of RV1 1630 are allocated to parity 1 bits 1610 and parity 2 bits 1620 in some suitable proportion. For example, approximately half of the remaining bit positions in RV1 1630 not taken by systematic bits 1650 may be allocated to a portion of
15 parity 1 bits 1610, while the other approximately half of the remaining bit positions may be allocated to a portion of parity 2 bits 1620.

20 **[0126]** The second redundancy version, RV2 1640, may include all remaining systematic bits 1660 that were not included in RV1 1630, as well as all remaining parity bits 1670 that were not included in RV1 1630. In particular, systematic bits 1660 includes the 25% of systematic bits 1600 that were not included in the first redundancy version, RV1 1630. Since the code rate of the
25 system is 0.5, systematic bits 1660 and parity bits 1670 may take up only half of the *M* bit positions available in RV2 1640. Therefore, the remaining bit positions 1680 in RV2 1640 are allocated to bits (e.g., systematic, parity 1, and/or parity 2 bits in some suitable proportion) that
30 were included in RV1 1630. In some embodiments, the bits

- 53 -

of RV2 1640 may be selected in accordance with process 1000 (FIG. 10), described above.

[0127] FIG. 17 illustrates the generation of redundancy versions 1730, 1740, and 1750 from systematic bits 1700, parity 1 bits 1710, and parity 2 bits 1720. The code rate of the system in this example is 1.0, and therefore all of systematic bits 1700 could be included in RV1 1730 if some of the systematic bits were not purposely partitioned to RV2 due to the *swap* value. With a *swap* value of 25% in this example, 75% of the bits in RV1 1730 are taken by systematic bits 1760. Systematic bits 1760 of RV1 1730 may be selected from systematic bits 1700 using any suitable puncturing algorithm. The remaining 25% of the bits in RV1 1730 are allocated to parity 1 bits 1710 and parity 2 bits 1720 in some suitable proportion.

[0128] The second redundancy version, RV2 1740, can include systematic bits 1770. Systematic bits 1770 may be those of systematic bits 1700 that were not selected for inclusion in the first redundancy version, RV1 1730. Thus, due to the code rate of 1.0, 25% of the bits in RV2 1740 may be taken by systematic bits 1770, while the remaining 75% of RV2 1740 may be allocated to parity bits 1780 that were not selected for inclusion in RV1 1730. In particular, parity bits 1780 may include parity 1 bits 1710 and parity 2 bits 1720 in some suitable proportion, where these parity 1 and parity 2 bits were not included as part of RV1 1730. Finally, due to the high code rate, in accordance with an embodiment, RV3 1750 may include only parity bits 1790, wherein

- 54 -

parity bits 1790 include all of the parity 1 bits 1710 and parity 2 bits 1720 that were not included in either RV1 1730 or RV2 1740.

[0129] In accordance with an embodiment of the present invention, a rate matcher (e.g., enhanced rate matcher 420 (FIG. 4) or enhanced rate matcher 540 (FIG. 5), etc.) can separately execute puncturing algorithms on the different data streams - systematic, parity 1, and parity 2 streams - to select bits for inclusion in a current redundancy version. For example, the rate matcher may loop through the systematic bits and selectively puncture a portion of the systematic bits such that the punctured bits are not included in the currently redundancy version. The rate matcher may then loop through the parity 1 bits, and then the parity 2 bits, in a similar fashion to selectively puncture bits.

[0130] FIG. 18 shows an illustrative flowchart of process 1800 for selecting systematic, parity 1, and parity 2 bits for inclusion in a current redundancy version, and generating the current redundancy version using the selected bits. The steps of process 1800 may be performed by any suitable rate matcher implemented on a transmitting station or a receiving station, for example, process 1800 begins at step 1801 and proceeds to step 1802. At step 1802, the rate matcher receives system-level rate matching parameters. The system-level rate matching parameters may be obtained from another component in the transmitting station or receiving station, and may be based on a particular transmission protocol. For example, the system-level rate matching

- 55 -

parameters may include any of the parameters shown for the transmission protocols in Table 2 of FIG. 13 (e.g., *swap*, N_w , M_{NO_PAN} , M_{PAN}). Other system-level rate matching parameters may include, for example, R_{MAX} and whether a
5 PAN field should be included in the current redundancy version.

[0131] Then, at step 1804, the rate matcher can identify the type of the current redundancy version. In particular, using the value of R_{MAX} , the rate matcher may
10 determine whether the current redundancy version is RV1, RV2 type I, RV2 type II, or RV3. Process 1800 may then proceed to one, two, or all three of steps 1806, 1810, and 1814. In particular, in some embodiments, the rate matcher may execute these steps concurrently or
15 substantially concurrently. In other embodiments, the rate matcher may execute steps 1806 and 1808 before moving to steps 1810 and 1812, and may then execute steps 1814 and 1816 once steps 1810 and 1812 are complete. That is, the rate matcher may use a serial
20 approach that takes more time to complete, but can use the same resources (e.g., hardware, circuitry, logic, etc.) to complete the computations at these steps.

[0132] At step 1806, the rate matcher may initialize a first set of loop parameters for use in puncturing the
25 systematic bits. That is, the loop parameters may be used to enable the rate matcher to determine which systematic bits to exclude from the current redundancy version. The rate matcher may compute the values of the first set of loop parameters based on the type of the
30 current redundancy version (as identified at step 1804),

- 56 -

using computations (e.g., loop parameter computation equations) specific to systematic bit puncturing, and using the system-level rate matching parameters (as received at step 1802). The first set of loop parameters
5 may include values for: *flip*, *e_{plus}*, *e_{minus}*, *e2_{plus}*, and *e2_{minus}*. The *flip* variable was described above in connection with FIGS. 12A-13. The remaining parameters will be described below in connection with FIGS. 19 and 20.

10 **[0133]** Then, at step 1808, the rate matcher may execute loop logic to puncture a subset of the systematic bits. The rate matcher may use the first set of loop parameters calculated at step 1806 as inputs into the loop logic. Using the loop logic, the rate matcher may
15 loop through each of the systematic bits to determine whether each bit should be included in the current redundancy version. The determination for each bit is based at least partially on the first set of loop parameters. Thus, at the conclusion of the loop logic,
20 the rate matcher may have punctured a subset of the systematic bits, leaving the remaining systematic bits to be included in the current redundancy version. Process 1800 may then proceed to step 1810 (for a serial implementation) or to step 1818 (for a parallel
25 implementation).

[0134] At step 1810, the rate matcher may initialize a second set of loop parameters for use in puncturing the parity 1 bits. That is, the loop parameters may be used
30 bits to exclude from the current redundancy version. The

- 57 -

rate matcher may compute the values of the second set of loop parameters based on the type of the current redundancy version (as identified at step 1804), using computations (e.g., loop parameter computation equations) specific to parity 1 bit puncturing, and using the system-level rate matching parameters (as received at step 1802). The second set of loop parameters may include values for: *flip*, *e_{ini}*, *e_{plus}*, *e_{minus}*, *e2_{ini}*, *e2_{plus}*, and *e2_{minus}*, and T. The *flip* variable was described above in connection with FIGS. 12A-13. The remaining parameters will be described below in connection with FIGS. 19 and 20.

[0135] Then, at step 1812, the rate matcher may execute the loop logic to puncture a subset of the parity 1 bits. The rate matcher may use the second set of loop parameters calculated at step 1810 as inputs into the loop logic. Using the loop logic, the rate matcher may loop through each of the parity 1 bits to determine whether each bit should be included in the current redundancy version. The determination for each bit is based at least partially on the second set of loop parameters. Thus, even though the rate matcher may use the same loop logic implementation to puncture parity 1 bits as that used to puncture the systematic bits at step 1808, the different loop parameters used as input into the loop logic may cause the rate matcher to select a different number of parity 1 bits and at different bit positions. The different inputs may even cause the loop logic to execute a different puncturing algorithm, since the different loop parameters used for the parity 1 bits

- 58 -

may cause the loop logic to branch off and execute a different portion of the logic. At the conclusion of the loop logic, the rate matcher may have punctured a subset of the parity 1 bits, leaving the remaining parity 1 bits
5 to be included in the current redundancy version.

Process 1800 may then proceed to step 1814 (for a serial implementation) or to step 1818 (for a parallel implementation).

[0136] At step 1814, the rate matcher may initialize a
10 third set of loop parameters for use in puncturing the parity 2 bits. That is, the loop parameters may be used to enable the rate matcher to determine which parity 2 bits to exclude from the current redundancy version. The rate matcher may compute the values of the third set of
15 loop parameters based on the type of the current redundancy version (as identified at step 1804), using computations (e.g., loop parameter computation equations) specific to parity 2 bit puncturing, and using the system-level rate matching parameters (as received at
20 step 1802). The third set of loop parameters may include values for: *flip*, *e_{ini}*, *e_{plus}*, *e_{minus}*, *e2_{ini}*, *e2_{plus}*, and *e2_{minus}*, and T. The third set of loop parameters may be the same as or different from the second set of loop parameters. The *flip* variable was described above in
25 connection with FIGS. 12A-13. The remaining parameters will be described below in connection with FIGS. 19 and 20.

[0137] Then, at step 1816, the rate matcher may execute the loop logic to puncture a subset of the
30 parity 2 bits. The rate matcher may use the third set of

- 59 -

loop parameters calculated at step 1814 as inputs into the loop logic. Using the loop logic, the rate matcher may loop through each of the parity 2 bits to determine whether each bit should be included in the current
5 redundancy version. The determination for each bit is based at least partially on the third set of loop parameters. Thus, even though the rate matcher may use the same loop logic implementation to puncture parity 2 bits as that used to puncture the systematic bits at
10 step 1808 or the parity 1 bits at step 1812, the different loop parameters used as input into the loop logic may cause the rate matcher to select a different number of parity 2 bits and/or bits at different bit positions. The different inputs may even cause the loop
15 logic to execute a different puncturing algorithm, since the different loop parameters used for the parity 2 bits may cause the loop logic to branch off and execute a different portion of the logic. At the conclusion of the loop logic, the rate matcher may have punctured a subset
20 of the parity 2 bits, leaving the remaining parity 2 bits to be included in the current redundancy version. Process 1800 may then proceed to step 1818.

[0138] With continued reference to FIG. 18, once the rate matcher reaches step 1818, the rate matcher may have
25 identified all of the bits, whether systematic, parity 1, or parity 2 bits, that are to be included in the current redundancy version. At step 1818, the rate matcher may generate a redundancy version that includes the punctured systematic, parity 1, and parity 2 bits. Process 1800 may
30 then continue to step 1820 and end.

- 60 -

[0139] FIG. 19 shows Table 3 with a subset of the loop parameters that are calculated for the first, second, and third sets of loop parameters. In particular, Table 3 illustrates the computations that the rate matcher may perform to obtain values for the loop parameters, e_{plus} , e_{minus} , $e2_{plus}$, and $e2_{minus}$. These computations are based on internal parameters (e.g., $X_{ir,3}$, etc.) that the rate matcher computes using system-level rate matching parameters.

10 [0140] The loop parameters illustrated in Table 3 are loop error correction values. Generally, the loop logic operates by puncturing bits such that a desired puncturing ratio is achieved. For example, if a rate matcher allocates 150 bits for parity 1 bits, and there are 300 parity 1 bits to choose from, the loop logic will attempt to achieve a puncturing ratio of 0.50. The rate matcher uses the error correction values to achieve the desired puncturing ratio. In particular, while looping through all of the parity 1 bits (for example), the rate

15 20 matcher can keep track of the difference between the current puncturing ratio and the desired puncturing ratio. Based on this difference, the rate matcher may incrementally adjust loop variable e by loop parameters $e2_{minus}$ and $e2_{plus}$ depending on whether a current bit has been punctured. The loop parameters $e2_{minus}$ and $e2_{plus}$ may be used to adjust the correction factors based on whether a PAN field is also being included. The $e2_{minus}$ and $e2_{plus}$ loop parameters will be described in greater detail below in connection with FIG. 23.

- 61 -

[0141] Since a different number of each type of bit (e.g., systematic, parity 1, and parity 2) are selected for each redundancy version, the desired puncturing ratio is different for each type of bit and for each type of redundancy version (e.g., RV1, RV2-I, RV2-II, and RV3). Thus, the internal parameters shown in Table 3 (FIG. 19) are computed using equations that are specific to systematic, parity 1, or parity 2 bits, and are also specific to the type of the current redundancy version. For example, in accordance with an embodiment, the rate matcher can compute the following internal parameters for use in generating the error values of a first redundancy version (RV1):

$$X_{i1,s} = N_w \quad (\text{EQ. 5})$$

$$X_{i1,p1} = N_{p1} \quad (\text{EQ. 6})$$

$$X_{i1,p2} = N_{p2} \quad (\text{EQ. 7})$$

$$N_{i1,sys} = \min(\lceil (1 - swap) \cdot N_w \rceil, M) \quad (\text{EQ. 8})$$

$$N_{i1,p1} = \left\lfloor \frac{M - N_{i1,w}}{2} \right\rfloor \quad (\text{EQ. 9})$$

$$N_{i1,p2} = \left\lfloor \frac{M - N_{i1,w}}{2} \right\rfloor \quad (\text{EQ. 10})$$

$$e_{ini} = \left\{ \left(X_{ir,1} - \left\lfloor (r-1) \cdot e_{plus} / r_{max} \right\rfloor - 1 \right) \bmod e_{plus} \right\} + 1 \quad (\text{EQ. 10b})$$

$$e2_{ini} = \left\{ \left(\tilde{X}_{ir,1} - \left\lfloor (r-1) \cdot e2_{plus} / r_{max} \right\rfloor - 1 \right) \bmod e2_{plus} \right\} + 1 \quad (\text{EQ. 10c})$$

25 In EQ. 6 and EQ. 7, N_{p1} and N_{p2} are equal to the number of parity 1 and parity 2 bits prior to rate matching,

- 62 -

respectively, which may be same or close to N_w . Thus, the X_{i1} variables show the initial number of systematic, parity 1, and parity 2 bits that are present prior to rate matching. The N_{t1} variables refer to the number of
5 bits of each type that are allocated for transmission in the first redundancy version. As shown in EQ. 8 through EQ. 10, all of the systematic bits except those swapped out of RV1 are included in the first redundancy version (if possible), while approximately half of the remaining
10 bits in the size M (e.g., M_{PAN} or M_{NO_PAN}) redundancy version are allocated to parity 1 bits, and the other approximately half of the remaining bits are allocated to parity 2 bits.

[0142] EQS. 5 through 10c are computed using the
15 system-level rate matching parameters defined by the transmission protocol used by the rate matcher. The values generated by these equations and any other loop parameters (e.g., *flip*) enable the rate matcher to select bits in accordance with processes 1400 (FIG. 14) and 1500
20 (FIG. 15), described above. The rate matcher may use other equations to compute loop parameters for the other redundancy versions (e.g., RV2-I, RV2-II, and RV3), which can be derived in a similar manner to EQS. 5 through 10c.

[0143] FIG. 20 shows an illustrative flowchart of a
25 process that includes loop logic 2000. Loop logic 2000 may be executed by a rate matcher (e.g., enhanced rate matcher 104 or 108 of FIG. 1) to select bits for inclusion in a current redundancy version. In some
30 embodiments, the rate matcher may execute loop logic 2000 to perform each of steps 1808, 1812, and 1816 of FIG. 18.

- 63 -

Therefore, the rate matcher may include two or three implementations of loop logic 2000 if any of steps 1808, 1812, or 1816 are performed in parallel. Otherwise, the rate matcher may include one implementation of loop logic
5 2000, and may initiate loop logic 2000 each time one of steps 1808, 1812, or 1816 is performed. For simplicity in describing the process, FIG. 20 will be described in terms of selecting bits from a plurality of data bits. It should be understood that these data bits may refer to
10 systematic bits, parity 1 bits, or parity 2 bits, depending on whichever step in FIG. 18 the rate matcher is currently performing.

[0144] The process of FIG. 20 begins at step 2002 and proceeds to step 2004. At step 2004, the rate matcher
15 may initialize a set of loop parameters. The rate matcher may compute an initial error value, e_{ini} , which is used in conjunction with other loop parameters to achieve a desired puncturing ratios. The rate matcher may then set the current error value, e , to this initial error
20 value. At step 2004, the rate matcher may also compute any of the loop parameters described above in connection with steps 1806, 1810, and 1814 (FIG. 18). Thus, the loop parameters may include any of the loop parameters described above in connection with FIGS. 18 and 19.
25 These loop parameters, as well as some of the system-level rate matching parameters, are sufficient to initiate and run loop logic 2000 to completion.

[0145] Then, at step 2006, the rate matcher may set a loop variable, m , to one. The loop variable is used to
30 index to a particular bit in the data stream. Thus, by

- 64 -

setting $m=1$, the rate matcher initializes loop logic 2000 to start the loop by considering the first bit in the data stream.

[0146] Continuing to step 2008, the rate matcher
5 determines whether the loop variable, m , is less than or equal to N_w . If the loop variable is greater than N_w , then loop logic 2000 has looped through all of the bits in the data stream. Therefore, loop logic 2000 ends at step 2010. Otherwise, loop logic 2000 continues to
10 step 2012 in which the rate matcher computes:

m th bit previously included xor not(*flip*)
(EXPR. 1)

If the rate matcher determines that this expression, EXPR. 1, is not equal to one, loop logic 2000 moves to
15 step 2014. At step 2014, the rate matcher determines whether the *flip* variable is equal to zero. If the *flip* variable is equal to zero, the rate matcher punctures the m th bit at step 2016 so that the m th bit is excluded from the current redundancy version. Loop logic 2000 then
20 moves to step 2028 to consider the next bit in the data stream. Otherwise, if the rate matcher determines that the *flip* variable is equal to one, loop logic 2000 moves directly to step 2028 without puncturing the m th bit, thereby including the m th bit in the current redundancy
25 version.

[0147] Returning to step 2012, if the rate matcher determines that EXPR. 1 equals one, loop logic 2000 instead continues to step 2018. At step 2018, the rate matcher adjusts the current error value, e , by
30 subtracting the current error value by e_{minus} . Then, at

- 65 -

step 2020, the rate matcher determines whether the current error value is less than zero. If not, the rate matcher does not puncture the mth bit, and loop logic 2000 moves to step 2028 to consider the next bit in the data stream. If, at step 2020, the rate matcher instead determines that the current error value is less than zero, this indicates that the mth bit should be punctured to maintain an appropriate error value. Thus, at step 2024, the rate matcher adjusts the current error value to indicate that the mth bit will be punctured. In particular, the rate matcher adds e_{plus} to the current error value. Then, at step 2026, the rate matcher punctures the mth bit so that the mth bit is not included in the current redundancy version. Loop logic 2000 then moves to step 2028 to consider the next bit in the data stream.

[0148] To consider the next bit, the rate matcher increments the loop variable, m, at step 2028. Loop logic 2000 may then return to step 2008, where the above-described steps are performed for the next bit. Once all of the bits are considered, the rate matcher will end at step 2010.

[0149] Loop logic 2000 of FIG. 20 illustrates an example of logic that the rate matcher may execute to generate redundancy versions in accordance with any of the previously described processes. In particular, even though different algorithms are used to select bits depending on the current type of data and redundancy version, loop logic 2000 is configured such that the initialized *flip* variable (among other loop parameters)

- 66 -

instructs the rate matcher to branch to particular steps in loop logic 2000 to execute the appropriate algorithm. Moreover, the values of e_{ini} , e_{minus} , and e_{plus} initialized at step 2004 are able to dictate how many and which of the bits are excluded from the current redundancy version so that the appropriate number of bits are included in the current redundancy version. Loop logic 2000, therefore, represents a concise and efficient way to implement the rate matcher's selection logic.

10 **[0150]** To show that loop logic 2000 generates redundancy versions in accordance with previously described processes (e.g., processes 600 (FIG. 6) and processes 700 (FIG. 7), etc.), FIGS. 22 and 23 illustrate what the rate matcher essentially accomplishes when performing loop logic 2000 for $R_{MAX} = 2$ and $R_{MAX} = 3$, respectively. In other words, FIGS. 22 and 23 illustrate the puncturing algorithm that loop logic 2000 implements for each type of redundancy version - e.g., RV1 and RV2-I for FIG. 22 and RV1, RV2-II, and RV3 for FIG. 23. The processes of FIGS. 22 and 23 may be derived from inserting some of the loop parameters (e.g., *flip*) into loop logic 2000 for each type of redundancy version.

20 **[0151]** Referring first to FIG. 21, a flowchart of illustrative process 2100 is shown that loop logic 2000 (FIG. 20) effectively implements when $R_{MAX} = 2$. Process 2100 begins at step 2102 and proceeds to step 2104. At step 2104, the rate matcher executing the loop logic may determine whether all bits (e.g., the systematic, parity 1, or parity 2 bits) have been considered. If all bits have been looped through,

30

- 67 -

process 2100 ends at step 2106. Otherwise, the rate
matcher moves to the next bit, m , and performs the
remaining steps of the FIG. 21 flowchart for this next
bit. If the current redundancy version is a first
5 redundancy version (2108), the rate matcher may
selectively puncture the m th bit based on the current
error value between the current and desired puncturing
ratios. Thus, the rate matcher can puncture a sufficient
number of bits so that the remaining, unpunctured bits
10 fit within the predetermined size of the redundancy
version (e.g., M). If the current redundancy version is
a type-I second redundancy version (2112), the rate
matcher may first determine at step 2114 whether the m th
bit has already been included in RV1. If the rate
15 matcher determines that the m th bit has not been included
in RV1, process 2100 moves back to step 2104 to consider
the next bit (if one exists). In this scenario, the rate
matcher does not puncture the m th bit, and therefore the
 m th bit is included in RV2 if it has not already been
20 included in RV1. This way, process 2100 (and therefore
loop logic 2000) ensures that all bits are included at
least once within the two redundancy versions.

[0152] If, at step 2114, the rate matcher determines
that the current m th bit has already been included in
RV1, process 2100 moves to step 2110, where the m th bit
25 may be punctured. Thus, any bit that has already been
included may or may not be selected for inclusion in the
second redundancy version, depending on whether
additional space is available in the M -bit redundancy

- 68 -

version. Process 2100 may then return to step 2104 to consider the next bit (if one exists).

[0153] Referring now to FIG. 22, a flowchart of illustrative process 2200 is shown that loop logic 2000 (FIG. 20) effectively implements when $R_{MAX} = 3$. Process 2200 begins at step 2202 and proceeds to step 2204. At step 2204, the rate matcher executing the loop logic may determine whether all bits (e.g., the systematic, parity 1, or parity 2 bits) have been considered. If all bits have been looped through, process 2200 ends at step 2206. Otherwise, the rate matcher moves to the next bit, m , and performs the remaining steps of the FIG. 22 flowchart for this next bit. If the current redundancy version is a first redundancy version (2208), the rate matcher may selectively puncture the m th bit based on the current error value between the current and desired puncturing ratios. Thus, the rate matcher can puncture a sufficient number of bits so that the remaining, unpunctured bits fit within the predetermined size of the redundancy version (e.g., M).

[0154] If the current redundancy version is a type-II second redundancy version (2212), the rate matcher behaves differently based on whether the data is systematic data or parity data. If the current redundancy version includes systematic data (2214), the rate matcher may first determine at step 2216 whether the m th systematic bit has already been included in $RV1$. If the rate matcher determines that the m th bit has not been included in $RV1$, process 2200 moves back to step 2204 to

- 69 -

consider the next bit (if one exists). In this scenario, the rate matcher does not puncture the *m*th systematic bit, and therefore the *m*th systematic bit is included in RV2 if it has not already been included in RV1. This way, process 2100 (and therefore loop logic 2000) ensures that all the systematic bits are included at least once within the two redundancy versions.

[0155] If, at step 2216, the rate matcher determines that the *m*th systematic bit has already been included in RV1, process 2200 moves to step 2210, where the *m*th bit may be punctured. Thus, any systematic bit that has already been included in RV1 may or may not be selected for inclusion in the second redundancy version, depending on whether additional space is available in the *M*-bit redundancy version. Process 2200 may then return to step 2204 to consider the next bit (if one exists).

[0156] Returning to step 2214, if the data bits are parity bits (e.g., a parity 1 or parity 2 data bits) and the rate matcher determines that the *m*th parity bit has already been included in RV1, the rate matcher punctures the current parity bit at step 2220. That is, the rate matcher automatically excludes any parity bit that has already been included in RV1 from being selected for inclusion in RV2. If the rate matcher determines that the current parity bit has not been included in RV1, the parity bit may or may not be punctured at step 2210 based on whether the *M*-bit second redundancy version has enough space available. Thus, the rate matcher selects, for inclusion in the second redundancy version, only a portion of the parity bits that were not included in RV1.

- 70 -

[0157] With the continued reference to FIG. 22, if the current redundancy version is a third redundancy version (2222), the rate matcher determines whether the current bit was included in RV1 or RV2 at step 2224. If the rate
5 matcher determines that the mth bit has not been included in either RV1 or RV2, process 2200 moves back to step 2204 to consider the next bit (if one exists). In this scenario, the rate matcher does not puncture the mth bit, and therefore the mth bit is included in RV3 if it has
10 not already been included in RV1 or RV2. This way, process 2100 (and therefore loop logic 2000) ensures that all of the bits are included in at least one of the three redundancy versions. Process 2200 then moves back to step 2204 to consider the next bit (if one exists).

15 [0158] Returning to step 2224, if the rate matcher determines that the mth bit has already been included in RV1 or RV2, process 2200 moves to step 2210, where the mth bit may be punctured. Thus, any bit that has already been included in RV1 or RV2 may or may not be selected
20 for inclusion in the third redundancy version, depending on whether additional space is available in the *M*-bit redundancy version. Process 2200 may then return to step 2204 to consider the next bit.

[0159] With continued reference to FIGS 20-22, loop
25 logic 2000 (FIG. 20) essentially illustrates one representative approach to implement the processes of FIGS. 21 and 22. It should be noted that the general selection strategy illustrated by the processes in FIGS. 21 and 22 may be implemented using any suitable set of
30 logic structures and loop variables.

- 71 -

[0160] Each packet of information transmitted by a transmitting station may or may not include a PAN field. Therefore, the number of bits available for each redundancy version may vary depending on whether a PAN field is included in the redundancy version. In some embodiments, loop logic 2000 of FIG. 20 represents the steps performed by a rate matcher when a PAN field is included in the transmitted packet (e.g., when $M = M_{PAN}$). Thus, if a PAN field is not included in the transmitted packet, additional systematic and/or parity bits may be selected for inclusion in a current redundancy version.

[0161] FIG. 23 shows an illustrative flowchart of loop logic 2300, which is similar to loop logic 2000 (FIG. 20), except that loop logic 2300 initializes a second error value, $e2 = e2_{ini}$, and includes inner PAN logic 2301. The second error value, $e2$, is used to obtain a desired puncturing ratio when a PAN field is not included, and therefore more space is available in the redundancy version. The second error value may be associated with correction factors, $e2_{plus}$ and $e2_{minus}$, which may be initialized and utilized in a similar manner as e_{plus} and e_{minus} described above. Inner PAN logic 2301 prevents some of the bits that loop logic 2000 (FIG. 20) would have punctured in order to accommodate a PAN field from being punctured when a PAN field is not included.

[0162] Inner PAN logic 2301 starts with step 2302 in which the rate matcher adjusts the second error value to maintain an accurate current error value. That is, the rate matcher subtracts $e2_{minus}$ from the current second error value. Then, at step 2304, the rate matcher

- 72 -

determines whether the second error value is less than or equal to zero. If not, the rate matcher does not puncture the mth bit, and loop logic 2300 moves out of inner PAN logic 2301 to consider the next bit. In this
5 scenario, inner PAN logic 2301 does not puncture a bit that loop logic 2000 (FIG. 20) would have punctured to accommodate a PAN field.

[0163] If, at step 2304, the rate matcher instead determines that the second error value is less than zero,
10 this indicates that the mth bit should be punctured to achieve the desired puncturing ratio. Thus, at step 2306, the rate matcher punctures the mth bit so that the mth bit is not included in the current redundancy version. In this scenario, the rate matcher punctures
15 the same bit that loop logic 2000 (FIG. 20) is configured to puncture. Then, at step 2026, since the current bit has been punctured, the rate matcher adjusts the second error value accordingly by adding e_{2plus} to the current second error value. Loop logic 2300 then moves out of
20 inner PAN logic 2301 to consider the next bit.

[0164] By implementing inner PAN logic, such as inner PAN logic 2301 of FIG. 23, a rate matcher can determine whether to puncture additional bits for the PAN field at the time of execution of loop logic 2300. In particular,
25 the possible inclusion of a PAN field does not affect the way in which e_{ini} , e_{plus} , and e_{minus} are computed. Thus, in an embodiment, the rate matcher may not have knowledge on whether a PAN field is to be included in the current radio block until the redundancy version for that radio
30 block is generated.

- 73 -

[0165] In some embodiments of the present invention, a rate matcher may distribute systematic bits between all three redundancy versions when $R_{MAX} = 3$, instead of including all of the systematic bits in the first two redundancy versions. This may be useful in systems with a high code rate - e.g., systems in which the code rate is at or near one, since few or no systematic bits would otherwise be included in the third redundancy version. By distributing all of the systematic bits between only the first two redundancy versions, the corresponding rate matcher may not be able to recover the transmitted message should RV1 or RV2 not be successfully received (e.g., if the header is not correctly decoded), especially because RV3 would not add any systematic bits to aid in decoding.

[0166] The rate matcher may distribute the systematic bits between the redundancy versions in a manner that improves decoding performance when combining pairs of received redundancy versions at a receiving station. In particular, the rate matcher allocates the systematic bits such that the combinations of RV1 and RV2, RV1 and RV3, as well as RV2 and RV3 can each be decoded successfully with high probability. Thus, the decoding performance may still be high in the event that a corresponding receiving station fails to receive one of the redundancy versions.

[0167] FIG. 24 shows an illustrative flowchart of process 2400 for generating three redundancy versions that each include a portion of the systematic bits. Process 2400 may be executed by any suitable transmitting

- 74 -

station or receiving stations implementing a rate
matcher. Process 2400 begins at step 2402 and proceeds
to step 2404. At step 2404, the rate matcher generates a
first redundancy version using a first puncturing
5 algorithm. The first redundancy version may have a
sufficient number of systematic bits to be decodable by
itself, but may include only a portion of all of the
systematic bits. The proportion of the systematic bits
included the first redundancy version may be based on a
10 *swap* value initialized prior to executing the first
puncturing algorithm.

[0168] At step 2408, the rate matcher may select a
second and a third subset of the systematic bits, where
each of the systematic bits not included in RV1 is
15 selected for at least one of the second and the third
subsets. The rate matcher may use the second subset of
systematic bits to generate a second redundancy version
at step 2410. The second redundancy version can include
the second subset of systematic bits. In an embodiment,
20 where the second subset may be selected such that the
expected decoding performance of RV1 and RV2 is improved.

[0169] Process 2400 of FIG. 24 may then continue to
step 2414. At step 2414, the rate matcher may generate a
third redundancy version that includes the third subset
25 of systematic bits selected previously at step 2408. The
third subset may have been selected such that the third
redundancy version generated at step 2414 is decodable in
combination with RV2 (without need of systematic bits
included only in RV1). In an embodiment the third
30 redundancy version has improved decoding performance when

- 75 -

combined with the first redundancy version. Process 2400 may then move to step 2418 and end.

[0170] FIG. 25 shows illustrative redundancy versions, RV1 2500, RV2 2510, and RV3 2520, generated in accordance with process 2400 of FIG. 24 in a system with a code rate of 1.0. RV1 2500 includes fewer than all of the systematic bits. In an embodiment, the remaining systematic bits 2530 are allocated to RV2 2510 and RV3 2520. The remaining bit positions in RV1 2500 are allocated to parity bits (e.g., parity 1 and/or parity 2 bits in some suitable proportion). RV1 2500 may include a sufficient number of systematic bits so that RV 2500 is still decodable by itself.

[0171] In some embodiments, RV1 2500, RV2 2510, and RV3 2520 may be generated such that all of the systematic bits and parity bits are included in the redundancy versions. Thus, due to code rate of one, each systematic bit and parity bit can only be included in one redundancy version. In these embodiments, the number of systematic bits selected for RV1 may be sufficient to allow a receiving station to decode RV1 by itself, but a number of the systematic bits may be reserved for transmission in RV2 2510 and RV3 2520 such that these two redundancy versions may be decodable together without systematic information from RV1.

[0172] In some embodiments of the present invention, a rate matcher may generate redundancy versions when $R_{MAX} = 3$ such that the third redundancy version may be the same as or similar to the first redundancy version. This approach may have relative computational simplicity, yet

- 76 -

may still produce sufficiently high decoding performance at the corresponding receiving station.

[0173] FIG. 26 shows an illustrative flowchart of process 2600 for generating three redundancy versions, where the third redundancy version is the same as the first redundancy version. Process 2600 may be executed by any suitable transmitting station or receiving station implementing a rate matcher. Process 2600 begins at step 2602 and proceeds to step 2604. At step 2604, the rate matcher generates a first redundancy version using a first puncturing algorithm and a first *swap* value (e.g., 5%, 15%, etc.). Thus, only a portion of the systematic bits defined by the *swap* value are selected by the first puncturing algorithm for inclusion in the first redundancy version. The rate matcher may then generate a first decodable redundancy version that includes the first subset of systematic bits at step 2606. The remaining bits of the first redundancy version may be allocated to parity bits (e.g., parity 1 and/or parity 2 bits in some suitable proportion).

[0174] Process 2600 then continues to step 2610, where the rate matcher selects a second subset of the systematic bits using a second puncturing algorithm. Since, in an embodiment, a distinct third redundancy version will not be generated, the rate matcher may include all of the systematic bits not selected for RV1 into the second subset of systematic bits. The rate matcher may then generate a second redundancy version at step 2612, which includes the second subset of systematic bits. The rate matcher may allocate the remaining bits

- 77 -

to parity bits, but may exclude any of the parity bits included in RV1 from being included in RV2 (e.g., by initializing the *flip* variable to zero). This maximizes the number of bits from the encoded message that are

5 included in the first two redundancy versions, since as noted additional bits will not be selected for the third redundancy version. If necessary, at step 2616, the rate matcher reuses the first redundancy version as the third redundancy version. Process 2600 may move to step 2618

10 and ends.

[0175] FIG. 27 shows illustrative redundancy versions, RV1 2700, RV2 2710, and RV3 2720, generated in accordance with process 2600 of FIG. 26 in a system with a code rate of 1.0. RV1 2700 includes a portion of the systematic

15 bits and a portion of the parity bits (e.g., parity 1 and/or parity 2 bits). RV2 2710 includes the systematic bits 2740 that are not included in RV1 2700, as well as parity 1 and/or parity 2 bits 2730 that are not included in RV1 2700. Therefore, in an embodiment maximum

20 percentage of the encoded message (e.g., 66% for $R=1$) is included within the first two redundancy versions. The third redundancy version, RV3 2720, is the same as RV1 2700, and therefore does not provide information on any of the remaining bits not included in RV1 or RV2.

25 However, including 66% of the encoded message, of which half is included in two redundancy versions, may be sufficient to obtain a high decoding performance at the corresponding receiving station.

[0176] In some embodiments of the present invention, a

30 rate matcher may generate redundancy versions in a manner

- 78 -

similar to that described in connection with FIGS. 26 and 27. However, rather than setting the third redundancy version to be the same as the first redundancy version, the rate matcher may generate the third
5 redundancy version using the same puncturing algorithm as the first redundancy version, but using a different *swap* value. For example, the rate matcher may use a *swap* value of 5% or 10% to generate the first redundancy version, and may use a *swap* value of 30% to generate the
10 third redundancy version. This technique simplifies the computational complexity of the rate matching approach for the third redundancy version compared to other approaches, since the third redundancy version is generated without consideration for which bits were
15 selected for the first and second redundancy versions. Yet, unlike the approach of reusing RV1 as RV3, this selection approach still allows different bits to be selected for the third redundancy version.

[0177] FIG. 28 shows an illustrative flowchart of
20 process 2800 for generating redundancy versions, where the first and third redundancy versions are generated using the same puncturing algorithm but different *swap* values. Process 2800 may be executed by any suitable transmitting or receiving station implementing a rate
25 matcher. Process 2800 begins at step 2802 and proceeds to step 2804. At step 2804, the rate matcher generates a first redundancy version using a first puncturing algorithm and a first *swap* value (e.g., 5%, 15%, etc.). Thus, only a portion of the systematic bits defined by
30 the *swap* value are selected by the first puncturing

- 79 -

algorithm for inclusion in the first redundancy version.
The rate matcher may then generate a first decodable
redundancy version that includes the first subset of
systematic bits at step 2806. The remaining bits of the
5 first redundancy version may be allocated to a first
subset of parity bits (e.g., parity 1 and/or parity 2
bits in some suitable proportion).

[0178] Process 2800 then continues to step 2810, where
the rate matcher selects a second subset of the
10 systematic bits using a second puncturing algorithm which
may be different from the first puncturing algorithm.
The rate matcher may select all of the systematic bits
not included in RV1 for the second subset of systematic
bits. The rate matcher may then generate a second
15 redundancy version at step 2812, which includes the
second subset of systematic bits. The rate matcher may
allocate the remaining bits to a second subset of parity
bits, but may exclude any of the parity bits included in
RV1 from being included in the second subset of parity
20 bits (e.g., by initializing the *flip* variable to zero).
This maximizes the number of bits from the encoded
message that are included in the first two redundancy
versions.

[0179] With continued reference to FIG. 28, at
25 step 2816, the rate matcher selects a third subset of the
systematic bits using the first puncturing and a second
swap value (e.g., 30%). The second *swap* value may be
different from the first *swap* value used to select the
first subset at step 2804. For example, the second *swap*
30 value may be greater than the first *swap* value so that a

- 80 -

different number of systematic bits at different bit positions are selected for inclusion in the third subset. Then, at step 2818, the rate matcher generates a third redundancy version that includes the third subset of systematic bits, and where the remaining bits in RV3 are allocated to a third subset of parity bits. Process 2800 may then move to step 2822 and end.

[0180] FIG. 29 shows illustrative redundancy versions, RV1 2900, RV2 2910, and RV3 2920, generated in accordance with process 2800 of FIG. 28 in a system with a code rate of 1.0. In this example, the *swap* value for RV1 2900 is initialized to 10% and the *swap* value for RV3 2920 is initialized to 30%. Therefore, RV1 2900 includes 90% systematic bits 2930, while RV3 2920 includes 70% systematic bits 2960. As described above, RV1 2900 and RV3 2920 are generated using the same puncturing algorithm. RV2 2910 may be generated using a different puncturing algorithm (or using the same puncturing algorithm with different loop parameters), since the rate matcher selects bits for inclusion in RV2 2910 based on which bits were selected for inclusion in RV1 2900. Thus, RV2 2910 may include systematic bits 2940, which is composed of all 10% of the systematic bits not selected for inclusion in RV1, and parity bits 2950, which is composed of parity 1 bits and parity 2 bits not selected for inclusion in RV1 2900.

[0181] The rate matcher may initialize the *flip* variable according to the values shown in Table 4 to generate redundancy versions in accordance with the technique described above in connection with FIGS. 28

- 81 -

and 29. The rate matcher may use these values regardless of whether the loop logic is being executed on a systematic, parity 1, or parity 2 data bits. In addition to the *flip* variable, the rate matcher may initialize
5 other loop parameters for use in puncturing systematic, parity 1, and parity 2 bits. The loop parameters may include any of the loop parameters described above, though some of the parameters may be computed differently due to the differences in the puncturing algorithms. The
10 rate matcher may use these loop parameters to separately execute loop logic on the data bits (e.g., using any of the techniques described above in connection with process 1800 of FIG. 18).

[0182] FIG. 31 shows an illustrative flowchart of a
15 process that includes loop logic 3100. Loop logic 3100 may be executed by a rate matcher (e.g., enhanced rate matcher 104 or enhanced rate matcher 108 of FIG. 1) to select bits for inclusion in a current redundancy version. In some embodiments, the rate matcher may
20 execute loop logic 3100 to perform each of steps 1808, 1812, and 1816 of FIG. 18. Therefore, the rate matcher may include two or three implementations of loop logic 3100 if any of steps 1808, 1812, or 1816 are performed in parallel. Otherwise, the rate matcher may
25 include one implementation of loop logic 3100, and may initiate loop logic 3100 each time one of steps 1808, 1812, or 1816 is performed. For simplicity in describing the process, FIG. 31 will be described in terms of selecting data bits. It should be understood that these
30 data bits may refer to systematic bits, parity 1 bits, or

- 82 -

parity 2 bits, depending on whichever step in FIG. 18 the rate matcher is currently performing.

[0183] The process of FIG. 31 begins at step 3102 and proceeds to step 3104. At step 3104, rate matcher may
5 initialize a set of loop parameters. The rate matcher may compute initial error value, e_{ini} and $e2_{ini}$, which are used in conjunction with other loop parameters to achieve a desired puncturing ratios. The rate matcher may then set the current error values, e and $e2$, to these initial
10 error values. The rate matcher may also initialize a binary loop vector referred to by the variable, $T(m)$, which includes N_w components. For the first and third redundancy versions, $T(m)$ may be initialized to all ones. For the second redundancy version, $T(m)$ may be a vector
15 that includes a one in all positions corresponding to bits were included in RV1 and zeros in all positions corresponding to bits that were punctured from RV1 (assuming that PAN was not included). This vector enables the rate matcher to identify which bits were
20 previously included in RV1, so that those bits will not be included in RV2.

[0184] Then, at step 3106, the rate matcher may set a loop variable, m , to one. The loop variable is used to index to a particular data bit. Thus, by setting $m=1$,
25 the rate matcher initializes loop logic 3100 to start the loop by considering the first bit.

[0185] Continuing to step 3108, the rate matcher determines whether the loop variable, m , is less than or equal to the N_w . If the loop variable is greater than N_w ,
30 then loop logic 3100 has looped through all of the bits.

- 83 -

Therefore, loop logic 3100 ends at step 3110. Otherwise, loop logic 3100 continues to step 3112 in which the rate matcher computes:

$$T(m) \text{ xor not}(\textit{flip}) \quad (\text{EXPR. 2})$$

5 If the rate matcher determines that this expression, EXPR. 2, is not equal to one, loop logic 3100 moves to step 3114. At step 3114, the rate matcher determines whether the *flip* variable is equal to zero. If the *flip* variable is equal to zero, the rate matcher punctures the
10 mth bit at step 3116 so that the mth bit is excluded from the current redundancy version. Loop logic 3100 then moves to step 3134 to consider the next bit. Otherwise, if the rate matcher determines that the *flip* variable is equal to one, loop logic 2000 moves directly to step 2034
15 without puncturing the mth bit. In this case, the mth bit is included the current redundancy version.

[0186] Returning to step 3112, if the rate matcher determines that EXPR. 2 equals one, loop logic 3100 instead continues to step 3118. At step 3118, the rate
20 matcher adjusts the current error value, e , by subtracting the current error value by e_{minus} . Then, at step 3120, the rate matcher determines whether the current error value is less than zero. If the current error value is less than zero, process 3100 continues to
25 step 3122, where the rate matcher punctures the mth bit. The rate matcher also adjusts the error value accordingly at step 3124 by adding e_{plus} to the current error value.

[0187] If, at step 3120, the rate matcher determines that the current error value is greater than zero, the
30 mth bit is not punctured unless additional bits are

- 84 -

needed to accommodate a PAN field. Loop logic 3100 therefore enters into inner PAN logic 3125, which determines whether the current bit should be punctured so that room for a PAN field can be made. Note that, unlike
5 in loop logic 2300, loop logic 3100 includes inner PAN logic that punctures additional bits if a PAN is included, rather than preventing bits from being punctured if a PAN is not included.

[0188] Inner PAN logic 3125 starts with step 3126 in
10 which the rate matcher adjusts the second error value to maintain an accurate current error value. That is, the rate matcher subtracts $e2_{minus}$ from the current second error value. Then, at step 3128, the rate matcher determines whether the second error value is less than or
15 equal to zero. If not, the rate matcher does not puncture the m th bit, and loop logic 2300 moves to step 3134 to consider the next bit.

[0189] If, at step 3128, the rate matcher instead determines that the second error value is less than zero,
20 this indicates that the m th bit should be punctured to achieve the desired puncturing ratio. Thus, at step 3130, the rate matcher punctures the m th bit so that the m th bit is not included in the current redundancy version. In this scenario, the rate matcher punctures
25 this bit so that a bit in the PAN field may be included in the current radio block. Then, at step 3132, since the current bit has been punctured, the rate matcher adjusts the second error value accordingly by adding $e2_{plus}$ to the current second error value. Loop logic 3100 then
30 moves to step 3134 to consider the next bit.

- 85 -

[0190] To show that loop logic 3100 generates redundancy versions in accordance with process 2800 of FIG. 28, FIG. 32 illustrates the effective process executed by a rate matcher when performing loop

5 logic 3100 for $R_{MAX} = 3$, respectively. In other words, FIG. 32 illustrates the puncturing algorithm that loop logic 3100 implements for each type of redundancy version - e.g., RV1, RV2-II, and RV3. The process of FIG. 32 may be derived from inserting some of the loop parameters

10 (e.g., the *flip* variables from Table 4 (FIG. 30)) into loop logic 3100 (FIG. 31) for each type of redundancy version. For $R_{MAX} = 2$, loop logic 3100 has a flow similar to that described above in connection with process 2100 of FIG. 21.

15 [0191] Process 3200 begins at step 3202 and proceeds to step 3204. At step 3204, the rate matcher executing the loop logic may determine whether all bits (e.g., the systematic, parity 1, or parity 2 bits) have been considered. If all bits have been looped through,

20 process 3200 ends at step 3206. Otherwise, the rate matcher moves to the next bit, m , and performs the remaining steps of the process 3200 for this next bit. If the current redundancy version is a first redundancy version (3208), the rate matcher may selectively puncture

25 the m th bit based on the current error value between the current and desired puncturing ratios. Thus, the rate matcher can puncture a sufficient number of bits so that the remaining, unpunctured bits fit within the predetermined size of the redundancy version (e.g., M).

- 86 -

[0192] If the current redundancy version is a second redundancy version (3212), the rate matcher may determine at step 3214 whether the mth bit was included in the first redundancy version (e.g., using $T(m)$). If the rate
5 matcher determines that the mth bit has already been included in RV1, the rate matcher punctures the mth bit at step 3214 so that this bit is not included in the second redundancy version. Otherwise, process 3200 moves to step 3210, where the mth bit is selectively punctured.
10 Thus, any bits that have not been included in RV1 may or may not be included in the second redundancy version, depending on the amount of space available in the current redundancy version. Process 3200 may then move back to step 3204 to consider the next bit.

15 [0193] With continued reference to FIG. 32, if the current redundancy version is a third redundancy version (3218), process 3200 moves directly to step 3210, rather than considering whether the bit has been included in the first or the second redundancy versions. At
20 step 3210, the rate matcher may or may not puncture the current bit. Note that the rate matcher behaves in a similar manner for RV1 (3208) and RV3 (3218). In particular, in both situations, process 3200 moves directly to step 3210 to determine whether the current
25 bit should be punctured. Since the case of RV1 (3208) may be associated with a different *swap* value than that of RV3 (3218), the result of the determination at step 3210 of whether the current bit should be punctured may be different. Thus, even though the same general
30 puncturing algorithm is used in these scenarios, the rate

- 87 -

matcher may select a different number of bits and different bit positions for inclusion in the respective redundancy versions.

[0194] Referring now to FIGS. 33A and 33B, an
5 illustrative flowchart of process 3300 is shown for decoding received redundancy versions generated in accordance with the rate matching techniques described above in connection with FIGS. 28-32. Process 3300 may be performed by a receiving station (e.g., mobile station
10 106 or base station 102 of FIG. 1).

[0195] Process 3300 starts at step 3302 and proceeds to step 3304. At step 3304, the receiving station receives a burst from a wireless communications network (e.g., a cellular network) in a current time slot. The
15 receiving station may determine whether a complete packet of information has been received at step 3306. For example, the receiving station may need to receive four bursts to obtain a full packet of information. Process 3300 may return to step 3304 if all bursts for the
20 current packet have not been received, and otherwise moves to step 3308. At step 3309, the receiving station decodes the header portion of the received packet.

[0196] If, at step 3309, the receiving station determines that the header cannot be decoded
25 successfully, the receiving station may not be able to interpret the information in the data portion of the packet. In this scenario, process 3300 ends at step 3310. Otherwise, if the receiving station successfully decodes the header portion, the receiving station can
30 process the BSNs within the data portion of the received

- 88 -

packet. Process 3300 therefore continues to the steps shown within region 3311, which are steps that the receiving station performs for each of the BSNs in the data portion.

- 5 **[0197]** The steps within region 3311 begin with step 3312. At step 3312, the receiving station may read the data portion of the received packet to obtain one of the BSNs, and at step 3314, the receiving station may deinterleave the obtained BSN. Process 3300 then
- 10 continues to step 3316 within region 3315. In some embodiments, the steps within this region may be performed by a rate matcher implemented on the receiving station (e.g., enhanced rate matcher 108 of FIG. 1 or enhanced rate matcher 353 of FIG. 3). At step 3316, the
- 15 receiving station identifies the redundancy version type (e.g., RV1, RV2, or RV3) that the BSN corresponds to. If the receiving station determines that the BSN is a first redundancy version, at step 3318, the receiving station may generate a puncturing sequence using the RV1
- 20 puncturing algorithm, as described above in connection with FIGS. 28-32. The puncturing sequence may be, for example, a size N_w binary vector indicating which bit positions of the mother code are included in the received BSN.
- 25 **[0198]** If, at step 3316, the receiving station instead determines that the received BSN is a second redundancy version, the receiving station may generate a puncturing sequence corresponding to a second redundancy version. The puncturing algorithm for generating RV2, however, may
- 30 need the puncturing sequence for RV1. Thus, the

- 89 -

receiving station first generates a puncturing sequence using the RV1 puncturing algorithm at step 3320. The receiving station may generate the RV1 puncturing sequence for the case where no PAN field is included, regardless of whether a PAN field was actually used in the previous RV1 transmission. By performing step 3320, the receiving station may obtain T, a loop parameter used in the RV2 puncturing algorithm (described above). Process 3300 may then continue to step 3322 in which the receiving station generates a puncturing sequence using the RV2 puncturing algorithm and the loop parameter, T, obtained from step 3320. The RV2 puncturing sequence may be, for example, a size N_w binary vector indicating which bit positions of the mother code are included in the received BSN.

[0199] Returning to step 3316, if the receiving station instead determines that the BSN is a third redundancy version, process 3300 continues to step 3324. At step 3324, the receiving station generates a puncturing sequence using the RV3 puncturing algorithm. For example, the receiving station may generate the puncturing sequence using the puncturing algorithm that would have been used for RV1 at step 3318, except the receiving station may use a different *swap* value. The RV3 puncturing sequence may be, for example, a size N_w binary vector indicating which bit positions of the mother code are included in the received BSN. In another embodiment (not shown), the receiving station may need to generate the puncturing sequences for RV1 and/or RV2 in order to generate the puncturing sequence for RV3.

- 90 -

[0200] Once the receiving station generates a puncturing sequence for the current redundancy version at step 3318, step 3322, or step 3324, process 3300 continues to step 3326. At step 3326, the receiving station, via its rate matcher, may unpuncture the de-interleaved BSN using the generated RV1, RV2, or RV3 puncturing sequence. In particular, the receiving station may map the bits of the de-interleaved BSN to particular bit positions of the mother code based on the puncturing sequence. For the remaining bit positions of the mother code, the receiving station may use erasures as placeholders.

[0201] Process 3300 may then continue to step 3328, which is shown in FIG. 33B. At step 3328, the receiving station determines whether previous versions of the same BSN has been received. If previous versions have been received, the receiving station may combine the unpunctured, current BSN to the previous versions. The receiving station may combine these redundancy versions using incremental redundancy combining, for example, or any other suitable combining technique. If, at step 3328, the receiving station instead determines that no previous versions of the same BSN have been received, process 3300 skips to step 3332 without performing combining.

[0202] At step 3332, the receiving station decodes the unpunctured BSN, which may or may not be combined with previously received versions of the same BSN. If, at step 3334, the receiving station determines that the unpunctured BSN has been successfully decoded,

- 91 -

process 3300 moves to step 3336. At step 3336, the receiving station passes the decoded data to the parts of the receiving station that use the decoded data, and process 3300 ends at step 3310. If, at step 3334, the
5 receiving station instead determines that decoding was unsuccessful, the receiving station may store the corresponding log-likelihood ratios (LLRs) of the BSN. The receiving station may then use the stored LLRs when the next redundancy version of the same BSN is received.
10 In particular, the receiving station can combine the stored LLRs with the next received BSN when the receiving station performs step 3330 for the next received version of the same BSN. Process 3300 may then end at step 3310.
[0203] The foregoing describes apparatuses and methods
15 for rate matching. Those skilled in the art will appreciate that the invention can be practiced by other than the described embodiments, which are presented for the purpose of illustration rather than of limitation.

- 92 -

What is Claimed is:

1. A method of transmitting a message over a wireless communications network, the method comprising:
 - encoding the message using a Turbo code to produce encoded data, the encoded data comprising a plurality of systematic bits and a plurality of parity bits;
 - selecting, for transmission over the wireless communications network, a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;
 - generating a first redundancy version of the message that has a predetermined number of bit positions, the first redundancy version including the first subset of the systematic bits, wherein unused bit positions of the first redundancy version are allocated to a first subset of the parity bits;
 - selecting, for transmission over the wireless communications network, a second subset of the systematic bits that has substantially all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;
 - generating a second redundancy version of the message that has the predetermined number of bit positions, the second redundancy version including the second subset of systematic bits, wherein unused bit positions of the second redundancy version are allocated to a second subset of parity bits that is generated from parity bits which are not included in the first subset of parity bits, and includes additional parity bits only if

- 93 -

30 all of the parity bits are otherwise included in the
first or second subsets; and
transmitting at least the first redundancy
version.

2. The method of claim 1, wherein selecting
the first subset or the second subset of systematic bits
comprises puncturing a portion of the systematic bits
such that the portion is excluded from the that subset.

3. The method of claim 1, further comprising:
selecting, for transmission over the
wireless communications network, a third subset of the
systematic bits using the first puncturing algorithm and
5 a second parameterized swap value, wherein the second
parameterized swap value is different from the first
parameterized swap value; and
generating a third redundancy version of
the message that has the predetermined number of bit
10 positions, the third redundancy version including the
third subset of the systematic bits and a third subset of
the parity bits.

4. The method of claim 3, wherein the second
parameterized swap value is greater than the first
parameterized swap value.

5. The method of claim 3, wherein the second
parameterized swap value is 0.30, and wherein selecting
the first subset of systematic bits comprises excluding
30% of the systematic bits from being included in the
5 first subset.

- 94 -

6. The method of claim 1, wherein the plurality of parity bits comprises a first plurality of parity bits and a second plurality of parity bits, the method further comprising:

5 allocating a portion of the first and second subsets of parity bits to the first plurality of parity bits; and

allocating a remaining portion of the first and second subsets of parity bits to the second
10 plurality of parity bits.

7. The method of claim 6, wherein approximately half of the first and second subsets of parity bits are allocated to the first plurality of parity bits, and a remaining approximately half are
5 allocated to the second plurality of parity bits.

8. The method of claim 3, wherein the plurality of parity bits comprises a first plurality of parity bits and a second plurality of parity bits, the method further comprising:

5 allocating a portion of the first, second, and third subsets of parity bits to the first plurality of parity bits; and

allocating a remaining portion of the first, second, and third subsets of parity bits to the
10 second plurality of parity bits.

9. The method of claim 1, wherein selecting the second subset of parity bits comprises:

- 95 -

excluding parity bits that are included in
the first subset of the parity bits from the second
5 subset of parity bits.

10. The method of claim 9, wherein excluding
the parity bits from being selected for the second subset
is performed in response to determining that more than
two redundancy versions are needed to transmit all bits
5 of the encoded data.

11. The method of claim 1, wherein including
all of the parity bits in the second subset that have not
been included in the first redundancy version is
performed in response to determining that two redundancy
5 versions are needed to transmit all bits of the encoded
data stream.

12. A method of receiving a message over a
wireless communications network, the method comprising:
receiving a first redundancy version of
the message that has a predetermined number of bit
5 positions, wherein the first redundancy version includes
a first subset of systematic bits and a first subset of
parity bits;
generating a first redundancy version
puncturing sequence based on a first puncturing algorithm
10 and a first parameterized swap value;
identifying the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
identifying the first subset of parity
15 bits, wherein bit positions of the first redundancy

- 96 -

version that are not identified as the first subset of systematic bits are identified as the first subset of the parity bits;

20 decoding the first subset of the
systematic and the parity bits to produce a first estimate of the message based on a Turbo code;

 receiving a second redundancy version of the message that has a predetermined number of bit positions, wherein the second redundancy version includes
25 a second subset of systematic bits and a second subset of parity bits;

 generating a second redundancy version puncturing sequence;

30 identifying the second subset of
systematic bits using the second redundancy version puncturing sequence;

 identifying the second subset of parity bits, wherein bit positions of the second redundancy version that are not identified as the second subset of
35 systematic bits are identified as the second subset of the parity bits;

 combining the first and second subset of systematic bits and the first and second subset of parity bits, wherein the combined subsets of systematic bits are
40 substantially all of the systematic bits corresponding to the message; and

 decoding the combined subsets of systematic and parity bits to produce a second estimate of the message based on a Turbo code.

- 97 -

13. The method of claim 12, further comprising:

receiving a third redundancy version of the message that has a predetermined number of bit positions and includes a third subset of systematic bits
5 and a third subset of parity bits;

generating a third redundancy version puncturing sequence based on the first puncturing algorithm and a second parameterized swap value, wherein
10 the second parameterized swap value is different from the first parameterized swap value; and

identifying the third subset of the systematic bits using the third redundancy version puncturing sequence.

14. The method of claim 13, wherein the second parameterized swap value is greater than the first second parameterized swap value.

15. The method of claim 13, wherein the second parameterized swap value is 0.30, and wherein identifying the first subset of systematic bits comprises determining that 30% of the systematic bits are not included in the
5 first subset of systematic bits.

16. The method of claim 12, wherein generating the second redundancy version puncturing sequence comprises generating the second redundancy version puncturing sequence based on the first redundancy version
5 puncturing sequence and a second puncturing algorithm.

- 98 -

17. The method of claim 12, wherein generating the second redundancy version puncturing sequence comprises:

generating another first redundancy
5 version puncturing sequence using the first puncturing algorithm and the first parameterized swap value, wherein the another first redundancy version puncturing sequence is different from the first redundancy version puncturing sequence; and
10 generating the second redundancy version puncturing sequence based on the another first redundancy version puncturing sequence and a second puncturing algorithm.

18. The method of claim 12, wherein generating the first or second redundancy version puncturing sequence comprises determining which bit positions of the systematic bits are included in the first or the second
5 subset of systematic bits.

19. The method of claim 12, wherein identifying the first or the second subset of the systematic bits or the parity bits comprises unpuncturing the first or second subset of the systematic bits or the
5 parity bits.

20. The method of claim 12, further comprising:

generating a first parity puncturing sequence, wherein the first subset of parity bits is
5 identified based on the first parity puncturing sequence.

- 99 -

21. The method of claim 12, wherein the first and second subset of parity bits each comprises a first plurality of parity bits and a second plurality of parity bits, the method further comprising:

5 generating a first parity puncturing sequence, wherein the first plurality of parity bits are identified based on the first parity puncturing sequence; and

 generating a second parity puncturing
10 sequence, wherein the second plurality of parity bits are identified based on the second parity puncturing sequence.

22. The method of claim 12, further comprising:

 determining that the decoding of the first subset of systematic bits and parity bits failed; and
5 receiving the second redundancy version responsive to the determining.

23. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

5 selecting a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;

 generating a first redundancy version of the message that has a predetermined number of bit
10 positions, the first redundancy version including the first subset of the systematic bits, wherein unused bit

- 100 -

positions of the first redundancy version are allocated to a first subset of the parity bits;

selecting a second subset of the
15 systematic bits that has substantially all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits; and
generating a second redundancy version of
the message that has the predetermined number of bit
20 positions, the second redundancy version including the second subset of systematic bits, wherein unused bit positions of the second redundancy version are allocated to a second subset of parity bits that is generated from parity bits which are not included in the first subset of
25 parity bits, and includes additional parity bits only if all of the parity bits are otherwise included in the first or second subsets.

24. The method of claim 23, wherein selecting the first subset or the second subset of systematic bits comprises puncturing a portion of the systematic bits such that the portion is excluded from the that subset.

25. The method of claim 23, further comprising:

selecting a third subset of the systematic
bits using the first puncturing algorithm and a second
5 parameterized swap value, wherein the second parameterized swap value is different from the first parameterized swap value; and

generating a third redundancy version of the message that has the predetermined number of bit

- 101 -

10 positions, the third redundancy version including the third subset of the systematic bits and a third subset of the parity bits.

26. The method of claim 25, wherein the second parameterized swap value is greater than the first second parameterized swap value.

27. The method of claim 25, wherein the second parameterized swap value is 0.30, and wherein selecting the first subset of systematic bits comprises excluding 30% of the systematic bits from being included in the
5 first subset.

28. The method of claim 23, wherein the plurality of parity bits comprises a first plurality of parity bits and a second plurality of parity bits, the method further comprising:

5 allocating a portion of the first and second subsets of parity bits to the first plurality of parity bits; and

allocating a remaining portion of the first and second subsets of parity bits to the second
10 plurality of parity bits.

29. The method of claim 28, wherein approximately half of the first and second subsets of parity bits are allocated to the first plurality of parity bits, and a remaining approximately half are
5 allocated to the second plurality of parity bits.

30. The method of claim 25, wherein the plurality of parity bits comprises a first plurality of

- 102 -

parity bits and a second plurality of parity bits, the method further comprising:

- 5 allocating a portion of the first, second, and third subsets of parity bits to the first plurality of parity bits; and
- allocating a remaining portion of the first, second, and third subsets of parity bits to the
- 10 second plurality of parity bits.

31. The method of claim 23, wherein selecting the second subset of parity bits comprises:

- excluding parity bits that are included in the first subset of the parity bits from the second
- 5 subset of parity bits.

32. The method of claim 31, wherein excluding the parity bits from being selected for the second subset is performed in response to determining that more than two redundancy versions are needed to include all bits of
- 5 the encoded data.

33. The method of claim 23, wherein including all of the parity bits in the second subset that have not been included in the first redundancy version is performed in response to determining that two redundancy
- 5 versions are needed to include all bits of the encoded data.

34. A base station for transmitting a message over a wireless communications network, the base station comprising:

- 103 -

a Turbo encoder configured to encode the
5 message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;

a rate matcher configured to:

select, for transmission over the
10 wireless communications network, a first subset of the
systematic bits using a first puncturing algorithm and a
first parameterized swap value;

generate a first redundancy version
that has a predetermined number of bit positions, the
15 first redundancy version including the first subset of
the systematic bits, wherein unused bit positions of the
first redundancy version are allocated to a first subset
of the parity bits;

select, for transmission over the
20 wireless communications network, a second subset of the
systematic bits that has substantially all of the
systematic bits corresponding to the message that are not
included in the first subset of the systematic bits; and

generate a second redundancy version
25 that has the predetermined number of bit positions, the
second redundancy version including the second subset of
the systematic bits, wherein unused bit positions of the
second redundancy version are allocated to a second
subset of parity bits that is generated from parity bits
30 which are not included in the first subset of parity
bits, and includes additional parity bits only if all of
the parity bits are otherwise included in the first or
second subsets; and

- 104 -

35 a transmitter configured to transmit at
least the first redundancy version over the wireless
communications network.

35. The base station of claim 34, wherein the
rate matcher is further configured to:

select the first subset or the second
subset of systematic bits by puncturing a portion of the
5 systematic bits such that the portion is excluded from
the that subset.

36. The base station of claim 34, wherein the
rate matcher is further configured to:

select, for transmission over the wireless
communications network, a third subset of the systematic
5 bits using the first puncturing algorithm and a second
parameterized swap value, wherein the second
parameterized swap value is different from the first
parameterized swap value; and

generate a third redundancy version of the
10 message that has the predetermined number of bit
positions, the third redundancy version including the
third subset of the systematic bits and a third subset of
the parity bits.

37. The base station of claim 36, wherein the
second parameterized swap value is greater than the first
parameterized swap value.

38. The base station of claim 36, wherein the
second parameterized swap value is 0.30, and wherein the
rate matcher is further configured to:

- 105 -

select the first subset of systematic bits
5 by excluding 30% of the systematic bits from being
included in the first subset.

39. The base station of claim 34, wherein the
plurality of parity bits comprises a first plurality of
parity bits and a second plurality of parity bits, the
rate matcher further configured to:

5 allocate a portion of the first and second
subsets of parity bits to the first plurality of parity
bits; and

allocate a remaining portion of the first
and second subsets of parity bits to the second plurality
10 of parity bits.

40. The base station of claim 39, wherein
approximately half of the first and second subsets of
parity bits are allocated to the first plurality of
parity bits, and a remaining approximately half are
5 allocated to the second plurality of parity bits.

41. The base station of claim 36, wherein the
plurality of parity bits comprises a first plurality of
parity bits and a second plurality of parity bits, the
rate matcher further configured to:

5 allocate a portion of the first, second,
and third subsets of parity bits to the first plurality
of parity bits; and

allocate a remaining portion of the first,
second, and third subsets of parity bits to the second
10 plurality of parity bits.

- 106 -

42. The base station of claim 34, wherein the rate matcher is further configured to:

select the second subset of parity bits by excluding parity bits that are included in the first
5 subset of the parity bits from the second subset of parity bits.

43. The base station of claim 42, wherein the rate matcher is configured to exclude the parity bits from being selected for the second subset in response to determining that more than two redundancy versions are
5 needed to transmit all bits of the encoded data.

44. The base station of claim 34, wherein the rate matcher is configured to include all of the parity bits in the second subset that have not been included in the first redundancy version in response to determining
5 that two redundancy versions are needed to transmit all bits of the encoded data stream.

45. A mobile station for receiving a message over a wireless communications network, the mobile station comprising:

a receiver configured to receive a first
5 and a second redundancy versions corresponding to the message over the wireless communications network, wherein the redundancy versions comprise a plurality of systematic bits and a plurality of parity bits;

a rate matcher configured to:
10 generate a first redundancy version puncturing sequence for a first subset of the systematic

- 107 -

bits included in the first redundancy version based on a first puncturing algorithm and a first parameterized swap value;

15 identify the first subset of the systematic bits using the first redundancy version puncturing sequence;

 identify a first subset of parity bits from the first redundancy version, wherein bit positions of the first redundancy version that are not identified as the first subset of systematic bits are identified as the first subset of the parity bits;

 generate a second redundancy version puncturing sequence for a second subset of the systematic bits included in the second redundancy version;

25 identify the second subset of systematic bits using the second redundancy version puncturing sequence;

 identify a second subset of parity bits from the second redundancy version, wherein bit positions of the second redundancy version that are not identified as the second subset of systematic bits are identified as the second subset of the parity bits; and

 a Turbo decoder configured to decode the first and the second subsets of the systematic and the parity bits to produce the message.

46. The mobile station of claim 45, wherein the rate matcher is further configured to:

 receive a third redundancy version of the message that has a predetermined number of bit positions

- 108 -

5 and includes a third subset of the systematic bits and a
third subset of the parity bits;
generate a third redundancy version
puncturing sequence based on the first puncturing
algorithm and a second parameterized swap value, wherein
10 the second parameterized swap value is different from the
first parameterized swap value; and
identify the third subset of the
systematic bits using the third redundancy version
puncturing sequence.

47. The mobile station of claim 46, wherein
the second parameterized swap value is greater than the
first second parameterized swap value.

48. The mobile station of claim 46, wherein
the second parameterized swap value is 0.30, and wherein
identifying the first subset of systematic bits comprises
determining that 30% of the systematic bits are not
5 included in the first subset of systematic bits.

49. The mobile station of claim 45, wherein
the rate matcher is configured to:
generate a second redundancy version
puncturing sequence based on the puncturing sequence for
5 the first redundancy version; and
identify the second subset of the
systematic bits using the second redundancy version
puncturing sequence.

50. The mobile station of claim 45, wherein
the rate matcher is configured to:

- 109 -

generate another first redundancy version
puncturing sequence, wherein the another first redundancy
5 version puncturing sequence is different from the
puncturing sequence for the first redundancy version;
generate a second redundancy version
puncturing sequence based on the another first redundancy
version puncturing sequence; and
10 identify the second subset of the
systematic bits using the second redundancy version
puncturing sequence.

51. The mobile station of claim 45, wherein
the rate matcher is configured to generate the puncturing
sequence for the at least first redundancy version by
determining which bit positions of the systematic bits
5 are included in the first subset of systematic bits.

52. A rate matcher for generating redundancy
versions of a message, wherein the message includes a
plurality of systematic bits and a plurality of parity
bits, the rate matcher configured to:
5 select a first subset of the systematic
bits using a first puncturing algorithm and a first
parameterized swap value;
generate a first redundancy version that
has a predetermined number of bit positions, the first
10 redundancy version including the first subset of the
systematic bits, wherein unused bit positions of the
first redundancy version are allocated to a first subset
of the parity bits;

- 110 -

select a second subset of the systematic
15 bits that has all of the systematic bits corresponding to
the message that are not included in the first subset of
the systematic bits; and

generate a second redundancy version that
has the predetermined number of bit portions, the second
20 redundancy version including the second subset of the
systematic bits, wherein unused bit positions of the
second redundancy version are allocated to a second
subset of parity bits that is limited to parity bits
which are not included in the first subset of parity
25 bits, and includes additional parity bits only if all of
the parity bit are otherwise included in the first or
second subsets.

53. The rate matcher of claim 52, wherein
selecting the first subset or the second subset of
systematic bits comprises puncturing a portion of the
systematic bits such that the portion is excluded from
5 the that subset.

54. The rate matcher of claim 52, further
configured to:

select, for transmission over the wireless
communications network, a third subset of the systematic
5 bits using the first puncturing algorithm and a second
parameterized swap value, wherein the second
parameterized swap value is different from the first
parameterized swap value; and

generate a third redundancy version of the
10 message that has the predetermined number of bit

- 111 -

positions, the third redundancy version including the third subset of the systematic bits and a third subset of the parity bits.

55. The rate matcher of claim 54, wherein the second parameterized swap value is greater than the first parameterized swap value.

56. The rate matcher of claim 54, wherein the second parameterized swap value is 0.30, and wherein selecting the first subset of systematic bits comprises excluding 30% of the systematic bits from being included
5 in the first subset.

57. The rate matcher of claim 52, wherein the plurality of parity bits comprises a first plurality of parity bits and a second plurality of parity bits, the rate matcher further configured to:

5 allocate a portion of the first and second subsets of parity bits to the first plurality of parity bits; and

allocate a remaining portion of the first and second subsets of parity bits to the second plurality
10 of parity bits.

58. The rate matcher of claim 57, wherein approximately half of the first and second subsets of parity bits are allocated to the first plurality of parity bits, and a remaining approximately half are
5 allocated to the second plurality of parity bits.

59. The rate matcher of claim 54, wherein the plurality of parity bits comprises a first plurality of

- 112 -

parity bits and a second plurality of parity bits, the rate matcher further configured to:

- 5 allocate a portion of the first, second, and third subsets of parity bits to the first plurality of parity bits; and
- allocate a remaining portion of the first, second, and third subsets of parity bits to the second
- 10 plurality of parity bits.

60. The rate matcher of claim 52, wherein selecting the second subset of parity bits comprises:

- excluding parity bits that are included in the first subset of the parity bits from the second
- 5 subset of parity bits.

61. The rate matcher of claim 60, wherein excluding the parity bits from being selected for the second subset is performed in response to determining that more than two redundancy versions are needed to

5 include all bits of the encoded data.

62. The rate matcher of claim 52, wherein including all of the parity bits in the second subset that have not been included in the first redundancy version is performed in response to determining that two
- 5 redundancy versions are needed to transmit all bits of the encoded data stream.

63. A wireless communications network for transmitting and receiving a message, the network comprising:

 a transmitting station comprising:

- 113 -

5 a Turbo encoder configured to encode
the message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;

 a rate matcher configured to:

10 select, for transmission over
the wireless communications network, a first subset of
the systematic bits using a first puncturing algorithm
and a first parameterized swap value;

 generate a first redundancy
15 version that has a predetermined number of bit positions,
the first redundancy version including the first subset
of the systematic bits, wherein unused bit positions of
the first redundancy version are allocated to a first
subset of the parity bits;

20 select, for transmission over
the wireless communications network, a second subset of
the systematic bits that has substantially all of the
systematic bits corresponding to the message that are not
included in the first subset of the systematic bits;

25 generate a second redundancy
version that has the predetermined number of bit
positions, the second redundancy version including the
second subset of the systematic bits, wherein unused bit
positions of the second redundancy version are allocated
30 to a second subset of parity bits that is generated from
parity bits which are not included in the first subset of
parity bits, and includes additional parity bits only if
all of the parity bits are otherwise included in the
first or second subsets;

- 114 -

35 a transmitter configured to transmit
the first and second redundancy versions over the
wireless communications network;
 a receiving station comprising:
 a receiver configured to receive a
40 first and a second redundancy versions corresponding to
the message over the wireless communications network,
wherein the redundancy versions comprise a plurality of
systematic bits and a plurality of parity bits;
 a rate matcher configured to:
45 generate a first redundancy
version puncturing sequence for a first subset of the
systematic bits included in the first redundancy version
based on a first puncturing algorithm and a first
parameterized swap value;
50 identify the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
 identify a first subset of
parity bits from the first redundancy version, wherein
55 bit positions of the first redundancy version that are
not identified as the first subset of systematic bits are
identified as the first subset of the parity bits;
 generate a second redundancy
version puncturing sequence for a second subset of the
60 systematic bits included in the second redundancy
version;
 identify the second subset of
systematic bits using the second redundancy version
puncturing sequence;

- 115 -

65 identify a second subset of
parity bits from the second redundancy version, wherein
bit positions of the second redundancy version that are
not identified as the second subset of systematic bits
are identified as the second subset of the parity bits;
70 and

 a Turbo decoder configured to decode
the first and the second subsets of the systematic and
the parity bits to produce the message.

64. A method of transmitting a message over a
wireless communications network, the method comprising:
 selecting a particular transmission scheme
from a plurality of available transmission schemes;
5 encoding the message using a Turbo code to
produce encoded data, the encoded data comprising a
plurality of systematic bits;
 generating a plurality of redundancy
versions of the message, wherein each of the plurality
10 of systematic bits is allocated to at least one of the
redundancy versions, and wherein each of the redundancy
versions includes less than all of the systematic bits;
and
 transmitting the plurality of redundancy
15 versions over the wireless communications network,
wherein at least one of the redundancy versions is
transmitted over a plurality of time slots defined by a
time division multiple access (TDMA) protocol, and
wherein at least one of the encoding, the generating, or
20 the transmitting is performed based at least in part on
the particular transmission scheme.

- 116 -

65. The method of claim 64, wherein selecting the particular transmission scheme comprises:

selecting a particular modulation scheme from a plurality of available modulation schemes, wherein
5 transmitting the plurality of redundancy versions comprises modulating the plurality of redundancy versions using the particular modulation scheme.

66. The method of claim 64, wherein selecting the particular transmission scheme comprises:

selecting a particular coding scheme from a plurality of available coding schemes, wherein encoding
5 the message comprises processing the message using the particular coding scheme.

67. The method of claim 66, wherein each of the available encoding schemes is defined by one or more of: a code rate that results in a same maximum number of redundancy versions as that of another available encoding
5 scheme, a code rate that results in a different maximum number of redundancy versions as that of another available encoding scheme, and a data packet size.

68. The method of claim 64, further comprising interleaving the encoded data into a plurality of bursts, wherein each of the bursts is transmitted over a different one of the time slots.

69. The method of claim 64, wherein the message is a first message and each of the redundancy versions of the first message is transmitted over a radio

- 117 -

block, wherein transmitting at least one of the
5 redundancy versions comprises:
transmitting at least one redundancy
version of the first message in a same radio block as
another redundancy version associated with a second
message.

70. The method of claim 64, wherein the
encoded data further comprises a plurality of parity
bits, the method further comprising:
5 distributing the parity bits between the
first and second redundancy versions such that none of
the parity bits are included in both the first and second
redundancy versions until all of the parity bits are
included in either the first or second redundancy
versions.

71. The method of claim 64, wherein generating
a first one of the redundancy versions comprises:
choosing a sufficient number of the
systematic bits that is less than all of the systematic
5 bits such that the first one of the redundancy versions
is decodable.

72. A method of receiving a message over a
wireless communications network, the method comprising:
receiving a plurality of redundancy
versions of the message over the wireless communications
5 network, wherein at least one of the redundancy versions
is received over a plurality of time slots defined by a
time division multiple access (TDMA) protocol;

- 118 -

generating a puncturing sequence for each
of the plurality of redundancy versions;
10 identifying, from each of the plurality of
redundancy versions, a subset of systematic bits
corresponding to the message using the puncturing
sequences, wherein each of the redundancy versions
includes less than all of the systematic bits; and
15 decoding the identified bits using a Turbo
code to produce the message, and wherein at least one of
the receiving, the identifying, or the decoding is
performed based at least in part on a particular
transmission scheme.

73. The method of claim 72, wherein receiving
the plurality of redundancy versions comprises
demodulating the plurality of redundancy versions using a
particular modulation scheme.

74. The method of claim 72, wherein decoding
the identified bits comprises processing the bits using a
particular decoding scheme.

75. The method of claim 74, wherein each of
the available decoding schemes is defined by one or more
of: a code rate that results in a same maximum number of
redundancy versions as that of another available encoding
5 scheme, a code rate that results in a different maximum
number of redundancy versions as that of another
available encoding scheme, and a data packet size.

76. The method of claim 72, further comprising
deinterleaving the plurality of redundancy versions from

- 119 -

a plurality of bursts, wherein each of the bursts is received over a different one of the time slots.

77. The method of claim 72, wherein the message is a first message and each of the redundancy versions of the first message is received over a radio block, wherein receiving at least one of the redundancy
5 versions comprises:

receiving at least one redundancy version of the first message in a same radio block as another redundancy version associated with a second message.

78. The method of claim 72, wherein the plurality of redundancy versions further comprise a plurality of parity bits, the method further comprising:
identifying the plurality of parity bits
5 corresponding to the message using the puncturing sequences for each of the plurality of redundancy versions.

79. The method of claim 72, wherein generating the puncturing sequences comprises determining which bit positions of the systematic bits are included in the subsets of systematic bits.

80. The method of claim 72, wherein identifying the subsets of the systematic bits comprises unpuncturing the subsets of systematic bits or parity bits.

81. A base station for transmitting a message over a wireless communications network, the base station comprising:

- 120 -

5 a configuration unit configured to select
a particular transmission scheme from a plurality of
available transmission schemes;

a Turbo encoder configured to encode the
message using a Turbo code to produce encoded data,
wherein the encoded data comprises a plurality of
10 systematic bits;

a rate matcher configured to generate a
plurality of redundancy versions of the message, wherein
each bit of the encoded data is allocated to at least one
of the redundancy versions, and wherein each of the
15 redundancy versions includes less than all of the
systematic bits; and

a transmitter configured to transmit the
plurality of redundancy versions over the wireless
communications network, wherein at least one of the
20 redundancy versions is transmitted over a plurality of
time slots defined by a time division multiple access
(TDMA) protocol, and wherein the at least one of the
Turbo encoder, the rate matcher, and the transmitter is
configured to operate based at least in part on the
25 particular transmission scheme.

82. A mobile station for receiving a message
over a wireless communications network, the mobile
station comprising:

a receiver configured to receive a
5 plurality of redundancy versions of the message over the
wireless communications network, wherein at least one of
the redundancy versions is received over a plurality of

- 121 -

time slots defined by a time division multiple access (TDMA) protocol;

10 a rate matcher configured to:

 generate a puncturing sequence for each the plurality of redundancy versions; and

 identify, from each of the plurality of redundancy versions, a subset of systematic bits

15 corresponding to the message using the puncturing sequences, wherein each of the redundancy versions includes less than all of the systematic bits; and

 a Turbo decoder configured to decode the identified bits using a Turbo code to produce the

20 message, and wherein at least one of the receiving, the identifying, or the decoding is performed based at least in part on a particular transmission scheme.

83. A wireless communications network for transmitting and receiving a message, the network comprising:

 a transmitting station comprising:

5 a configuration unit configured to select a particular transmission scheme from a plurality of available transmission schemes;

 a Turbo encoder configured to encode the message using a Turbo code to produce encoded data,

10 wherein the encoded data comprises a plurality of systematic bits;

 a rate matcher configured to generate a plurality of redundancy versions of the message, wherein each bit of the encoded data is allocated to at

15 least one of the redundancy versions, and wherein each of

- 122 -

the redundancy versions includes less than all of the systematic bits;

20 a transmitter configured to transmit the plurality of redundancy versions over the wireless communications network, wherein at least one of the redundancy versions is transmitted over a plurality of time slots defined by a time division multiple access (TDMA) protocol, and wherein the at least one of the Turbo encoder, the rate matcher, and the transmitter is
25 configured to operate based at least in part on the particular transmission scheme;

a receiving station comprising:

a receiver configured to receive a plurality of redundancy versions of the message over the
30 wireless communications network, wherein at least one of the redundancy versions is received over a plurality of time slots defined by a time division multiple access (TDMA) protocol;

a rate matcher configured to:

35 generate a puncturing sequence for each the plurality of redundancy versions; and

identify, from each of the plurality of redundancy versions, a subset of systematic bits corresponding to the message using the puncturing
40 sequences, wherein each of the redundancy versions includes less than all of the systematic bits; and

a Turbo decoder configured to decode the identified bits using a Turbo code to produce the message, and wherein at least one of the receiving, the

- 123 -

45 identifying, or the decoding is performed based at least
in part on the particular transmission scheme.

84. A method of transmitting a message over a
wireless communications network, the method comprising:

encoding the message using a Turbo code to
produce encoded data, the encoded data comprising a
5 plurality of systematic bits and a plurality of parity
bits;

selecting a first subset of the systematic
bits for transmission over the wireless communications
network, wherein the first subset of the systematic bits
10 includes less than all of the plurality of systematic
bits;

generating a first decodable redundancy
version of the message that includes the first subset of
the systematic bits and a first subset of parity bits;

15 selecting a second subset of the
systematic bits for transmission over the wireless
communications network, the second subset of the
systematic bits including all of the systematic bits
corresponding to the message that are not included in the
20 first subset of the systematic bits;

generating at least one additional
redundancy version of the message that includes the
second subset of the systematic bits and a second subset
of the parity bits, wherein the second subset of parity
25 bits are different from the first subset of parity bits;
and

transmitting at least the first redundancy
version.

- 124 -

85. The method of claim 84, the method further comprising:

initializing a swap value that defines a proportion of the systematic bits that are to be included
5 in the first subset; and
selecting the first subset based on the swap value.

86. The method of claim 84, wherein selecting the first or the second subset of systematic bits comprises:

puncturing a portion of the systematic
5 bits using a puncturing algorithm such that the portion is excluded from the first subset or the second subset.

87. The method of claim 84, wherein selecting the second subset comprises:

for each of the systematic bits:
determining whether the systematic bit has
5 been included in the first redundancy version; and
including the systematic bit in the second subset when the systematic bit has not been included in the first redundancy version.

88. The method of claim 87, wherein selecting the second subset further comprises:

selectively puncturing the systematic bit when the systematic bit has been included in the first
5 redundancy version.

89. The method of claim 84, wherein encoding the message further comprises:

- 125 -

producing the plurality of systematic bits, a first plurality of parity bits, and a second
5 plurality of parity bits.

90. The method of claim 89, wherein the first redundancy version has a predetermined number of bit positions, the method further comprising:

allocating the unused bit positions of the
5 first redundancy version to a combination of parity bits from the first plurality of parity bits and the second plurality of parity bits.

91. The method of claim 84, wherein transmitting at least the first redundancy version comprises:

transmitting at least the first redundancy
5 version over a plurality of time slots defined by a time division multiple access (TDMA) protocol.

92. The method of claim 89, further comprising interleaving at least the first redundancy version into a plurality of bursts, wherein each of the bursts is transmitted over a different one of the time slots.

93. The method of claim 84, further comprising:

determining whether to include a piggy-backed ACK/NACK (PAN) field in the first or the at least
5 one additional redundancy version.

94. The method of claim 93, wherein selecting the first or the second subset of systematic bits comprises:

- 126 -

puncturing a first portion of the
5 systematic bits such that the first portion is excluded
from the first subset or the second subset; and
in response to determining that a PAN
field should be transmitted in the first or the at least
one additional redundancy version, puncturing an
10 additional portion of the systematic bits such that the
additional portion is also excluded from the first subset
or the second subset.

95. The method of claim 84, further
comprising:

receiving a negative acknowledgement using
a hybrid automatic repeat request (HARQ) protocol,
5 wherein the at least one additional redundancy version is
transmitted in response to receiving the negative
acknowledgment.

96. A method of receiving a message over a
wireless communications network, the method comprising:

receiving a first decodable redundancy
version of the message, wherein the first redundancy
5 version includes a first subset of systematic bits and a
first subset of parity bits;
generating a first redundancy version
puncturing sequence for the first subset of systematic
bits, wherein the first redundancy version puncturing
10 sequence indicates that less than all of the systematic
bits are included in the first subset of systematic bits;

- 127 -

identifying the first subset of the systematic bits using the first redundancy version puncturing sequence;

15 decoding the identified first subset of the systematic bits and the first subset of parity bits to produce a first estimate of the message based on a Turbo code;

 receiving at least one additional
20 redundancy version of the message, wherein the at least one additional redundancy version includes a second subset of the systematic bits and a second subset of the parity bits;

 generating at least a second redundancy
25 version puncturing sequence for the second subset of systematic bits, wherein the at least a second redundancy version puncturing sequence indicates that the second subset of the systematic bits includes systematic bits corresponding to the message that were not identified in
30 the first subset of the systematic bits;

 identifying, from the least one additional redundancy version of the message, the second subset of the systematic bits using the at least a second redundancy version puncturing sequence;

35 combining the first and second subset of systematic bits and the first and second subset of parity bits; and

 decoding the combined subsets of systematic and parity bits to produce a second estimate
40 of the message based on a Turbo code.

- 128 -

97. The method of claim 96, the method further comprising:

determining a swap value that defines a proportion of the systematic bits that are included in
5 the first subset of the systematic bits; and
identifying the first subset of the systematic bits based on the swap value.

98. The method of claim 96, wherein the plurality of parity bits comprise a first plurality of parity bits and a second plurality of parity bits.

99. The method of claim 98, wherein the first redundancy version has a predetermined number of bit positions, the method further comprising:

identifying a combination of parity bits
5 from the first plurality and the second plurality within bit positions of the first redundancy version that are not identified as the first subset of systematic bits.

100. The method of claim 96, wherein generating the first or the at least a second redundancy version puncturing sequence comprises determining which bit positions of the systematic bits are included in the
5 first subset or the second subset of systematic bits.

101. The method of claim 96, wherein identifying the first or the second subset of systematic bits comprises unpuncturing the first or second subset of the systematic bits.

- 129 -

102. The method of claim 96, wherein receiving at least the first redundancy version comprises:

receiving at least the first redundancy version over a plurality of time slots defined by a time
5 division multiple access (TDMA) protocol.

103. The method of claim 102, further comprising deinterleaving at least the first redundancy version from a plurality of bursts, wherein each of the bursts is received over a different one of the time
5 slots.

104. The method of claim 96, further comprising:

determining whether a piggy-backed ACK/NACK (PAN) field is included in the first or the at
5 least one additional redundancy version.

105. The method of claim 104, wherein identifying the first or the second subset of systematic bits further comprises:

in response to determining that a PAN
5 field is included in the first or the at least one additional redundancy version, modifying the associated redundancy version puncturing sequence to indicate that the first or the second subset of systematic bits include fewer of the systematic bits.

106. The method of claim 96, further comprising:

determining that the decoding of the first subset of systematic bits and parity bits failed; and

- 130 -

5 in response to the determining,
transmitting a negative acknowledgement using a hybrid
automatic repeat request (HARQ) protocol, wherein the at
least one additional redundancy version is received in
response to the negative acknowledgment.

107. A method of generating redundancy versions
of a message, wherein the message includes a plurality of
systematic bits and a plurality of parity bits, the
method comprising:

5 selecting a first subset of the systematic
bits, wherein the first subset of the systematic bits
includes less than all of the plurality of systematic
bits;

 generating a first decodable redundancy
10 version of the message that includes the first subset of
the systematic bits and a first subset of parity bits;

 selecting a second subset of the
systematic bits, the second subset of the systematic bits
including all of the systematic bits corresponding to the
15 message that are not included in the first subset of the
systematic bits; and

 generating at least one additional
redundancy version of the message that includes the
second subset of the systematic bits and a second subset
20 of the parity bits that are different from the first
subset of parity bits.

108. A base station for transmitting a message
over a wireless communications network, the base station
comprising:

- 131 -

5 a Turbo encoder configured to encode the message to produce encoded data, the encoded data comprising a plurality of systematic bits and a first subset of parity bits;

a rate matcher configured to:

10 select a first subset of the systematic bits for transmission over the wireless communications network wherein the first subset of the systematic bits includes less than all of the plurality of systematic bits;

15 generate a first decodable redundancy version that includes the first subset of the systematic bits and a first subset of parity bits;

20 select a second subset of the systematic bits for transmission over the wireless communications network, the second subset of the systematic bits having all of the systematic bits that are not included in the first redundancy version; and

25 generate at least one additional redundancy version that includes the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of parity bits; and

a transmitter configured to transmit the first and the at least one additional redundancy versions over the wireless communications network.

109. A mobile station for receiving a message over a wireless communications network, the mobile station comprising:

- 132 -

5 a receiver configured to receive a first
decodable redundancy version of the message and at least
one additional redundancy version of the message, wherein
each of the redundancy versions includes a subset of
systematic bits and a subset of parity bits;
a rate matcher configured to:
10 generate a first redundancy version
puncturing sequence for a first subset of systematic bits
included in the first redundancy version, wherein the
first redundancy version puncturing sequence indicates
that less than all of the systematic bits are included in
15 the first subset of systematic bits;
identify the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
generate at least a second redundancy
20 version puncturing sequence for a second subset of
systematic bits included in the at least one additional
redundancy version, wherein the at least a second
redundancy version puncturing sequence indicates that the
second subset of the systematic bits includes systematic
25 bits corresponding to the message that were not
identified in the first subset of the systematic bits;
identify the second subset of the
systematic bits using the at least a second redundancy
version puncturing sequence;
30 a Turbo decoder configured to decode the
identified bits to produce the message.

110. A rate matcher for generating redundancy
versions of a message, wherein the message includes a

- 133 -

plurality of systematic bits and a plurality of parity bits, the rate matcher configured to:

5 select a first subset of the systematic bits, wherein the first subset of the systematic bits includes less than all of the plurality of systematic bits;

 generate a first decodable redundancy
10 version that includes the first subset of the systematic bits and a first subset of parity bits;

 select a second subset of the systematic bits, the second subset of the systematic bits having all of the systematic bits that are not included in the first
15 redundancy version; and

 generate at least one additional redundancy version that includes the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of parity bits.

111. A wireless communications network for transmitting and receiving a message, the network comprising:

 a transmitting station comprising:
5 a Turbo encoder configured to encode the message to produce encoded data, the encoded data comprising a plurality of systematic bits and a first subset of parity bits;

 a rate matcher configured to:
10 select a first subset of the systematic bits for transmission over the wireless communications network wherein the first subset of the

- 134 -

systematic bits includes less than all of the plurality of systematic bits;

- 15 generate a first decodable redundancy version that includes the first subset of the systematic bits and a first subset of parity bits;
- select a second subset of the systematic bits for transmission over the wireless
- 20 communications network, the second subset of the systematic bits having all of the systematic bits that are not included in the first redundancy version; and
- generate at least one additional redundancy version that includes the second subset of the
- 25 systematic bits and a second subset of the parity bits that are different from the first subset of parity bits;
- a transmitter configured to transmit the first and the at least one additional redundancy versions over the wireless communications network;
- 30 a receiving station comprising:
- a receiver configured to receive a first decodable redundancy version of the message and at least one additional redundancy version of the message, wherein each of the redundancy versions includes a subset
- 35 of systematic bits and a subset of parity bits;
- a rate matcher configured to:
- generate a first redundancy version puncturing sequence for a first subset of systematic bits included in the first redundancy version,
- 40 wherein the first redundancy version puncturing sequence indicates that less than all of the systematic bits are included in the first subset of systematic bits;

- 135 -

identify the first subset of the systematic bits using the first redundancy version puncturing sequence;

generate at least a second redundancy version puncturing sequence for a second subset of systematic bits included in the at least one additional redundancy version, wherein the at least a second redundancy version puncturing sequence indicates that the second subset of the systematic bits includes systematic bits corresponding to the message that were not identified in the first subset of the systematic bits;

identify the second subset of the systematic bits using the at least a second redundancy version puncturing sequence;

a Turbo decoder configured to decode the identified bits to produce the message.

112. A method of transmitting a message over a wireless communications network, the method comprising:

encoding the message using a Turbo code to produce encoded data, the encoded data comprising a plurality of systematic bits and a plurality of parity bits;

selecting a first subset of the parity bits for transmission over the wireless communications network;

generating a first redundancy version of the message that includes the first subset of the parity bits and a first subset of the systematic bits; and

- 136 -

selecting a second subset of the parity
bits for transmission over the wireless communications
15 network, wherein selecting the second subset comprises
excluding the parity bits selected for the first subset
from being selected for the second subset;

generating a second redundancy version of
the message that includes the second subset of the parity
20 bits and a second subset of the systematic bits.

113. The method of claim 112, further
comprising:

initializing a flip variable for each
redundancy version, wherein the flip variable is
5 initialized for the second redundancy version such that
the parity bits included in the first redundancy version
are excluded from the second redundancy version.

114. The method of claim 112, wherein selecting
the first or the second subset of parity bits comprises:

puncturing a portion of the parity bits
using a puncturing algorithm such that the portion is not
5 included in the first subset or the second subset.

115. The method of claim 112, further
comprising:

generating a third redundancy version of
the message in response to identifying that a code rate
5 of the Turbo code is greater than 0.66.

116. The method of claim 112, wherein selecting
the second subset comprises:

for each of the parity bits:

- 137 -

determining whether the parity bit has
5 been selected for the first subset; and
puncturing the parity bit when the parity
bit has been selected for the first subset.

117. The method of claim 116, wherein selecting
the second subset further comprises:
selectively puncturing the parity bit when
the parity bit has not been selected for the first
5 subset.

118. The method of claim 112, further
comprising including parity bits selected for the first
subset in the second subset once all of the plurality of
parity bits are otherwise included in the first or second
5 subsets.

119. The method of claim 112, further
comprising:
generating a third redundancy version that
includes all of the parity bits that have not been
5 included in the first redundancy version or the second
redundancy version.

120. The method of claim 112, further
comprising transmitting the first or the second
redundancy version over a plurality of time slots defined
by a time division multiple access (TDMA) protocol.

121. The method of claim 120, further
comprising interleaving the encoded data into a plurality
of bursts, wherein each of the bursts is transmitted over
a different one of the time slots.

- 138 -

122. The method of claim 112, further comprising:

5 determining whether to transmit a piggy-backed ACK/NACK (PAN) field in the first or the second redundancy versions.

123. The method of claim 122, wherein selecting the first or the second subset of parity bits comprises:

5 puncturing a first portion of the parity bits such that the first portion is excluded from the first subset or the second subset; and

10 in response to determining that a PAN field should be transmitted in the first or the second redundancy versions, puncturing an additional portion of the parity bits such that the additional portion is also excluded from the first subset or the second subset.

124. The method of claim 112, further comprising:

5 receiving a negative acknowledgement using a hybrid automatic repeat request (HARQ) protocol, wherein the second redundancy version is transmitted in response to receiving the negative acknowledgment.

125. A method of receiving a message over a wireless communications network, the method comprising:

5 receiving a first redundancy version of the message, wherein the first redundancy version includes a first subset of systematic bits and a first subset of parity bits;

- 139 -

generating a first redundancy version
puncturing sequence for the first subset of the parity
bits;
10 identifying the first subset of the parity
bits using the first redundancy version puncturing
sequence;
decoding the first subset of the
systematic bits and the identified first subset of the
15 parity bits to produce a first estimate of the message
based on a Turbo code;
receiving a second redundancy version of
the message, wherein the second redundancy version
includes a second subset of systematic bits and a second
20 subset of the parity bits;
generating a second redundancy version
puncturing sequence for the second subset of parity bits,
wherein the second redundancy version puncturing sequence
indicates that bit positions of the first subset of
25 parity bits do not correspond to parity bits in the
second subset;
identifying, from the second redundancy
version, the second subset of the parity bits using the
second redundancy version puncturing sequence;
30 combining the first and second subset of
systematic bits and the first and second subset of parity
bits; and
decoding the combined subsets of
systematic and parity bits to produce a second estimate
35 of the message based on a Turbo code.

- 140 -

126. The method of claim 125, further comprising receiving a third redundancy version of the message.

127. The method of claim 126, further comprising:

identifying, within the third redundancy version, all of the parity bits that have not been
5 identified in the first redundancy version or the second redundancy version.

128. The method of claim 125, wherein generating the first or the second redundancy version puncturing sequence comprises determining which bit positions of the parity bits are included in the first
5 subset or the second subset of the parity bits.

129. The method of claim 125, wherein identifying the first or the second subset of the parity bits comprises unpuncturing the first or second subset of parity bits.

130. The method of claim 125, further comprising receiving the first or the second redundancy version over a plurality of time slots defined by a time division multiple access (TDMA) protocol.

131. The method of claim 130, further comprising deinterleaving the redundancy versions from a plurality of bursts, wherein each of the bursts is transmitted over a different one of the time slots.

- 141 -

132. The method of claim 125, further comprising:

5 determining whether a piggy-backed ACK/NACK (PAN) field is included in the first or the second redundancy versions.

133. The method of claim 132, wherein identifying the first or the second subset of parity bits comprises:

5 in response to determining that a PAN field is included in the first or the second redundancy versions, modifying the associated redundancy version puncturing sequence to indicate that the first or the second subset of parity bits include fewer of the parity bits.

134. The method of claim 125, further comprising:

5 determining that the decoding of the first subset of systematic bits and parity bits failed; and in response to the determining, transmitting a negative acknowledgement using a hybrid automatic repeat request (HARQ) protocol, wherein the second redundancy version is received in response to the negative acknowledgment.

135. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

- 142 -

5 selecting a first subset of the parity
bits;
 generating a first redundancy version of
the message that includes the first subset of the parity
bits and a first subset of the systematic bits; and
10 selecting a second subset of the parity
bits, wherein selecting the second subset comprises
excluding the parity bits selected for the first subset
from being selected for the second subset; and
 generating a second redundancy version of
15 the message that includes the second subset of the parity
bits and a second subset of the systematic bits.

136. A base station for transmitting a message
over a wireless communications network, the base station
comprising:

 a Turbo encoder configured to encode the
5 message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;
 a rate matcher configured to:
 select a first subset of the parity
10 bits for transmission over the wireless communications
network;
 generate a first redundancy version
of the message, the first redundancy version including
the first subset of the parity bits and a first subset of
15 the systematic bits;
 select a second subset of the parity
bits for transmission over the wireless communications
network, wherein the parity bits that are selected for

- 143 -

the first subset are excluded from being selected for the
20 second subset; and

generate a second redundancy version
of the message, the second redundancy version including
the second subset of the parity bits and a first subset
of the systematic bits; and

25 a transmitter configured to transmit the
first and the second redundancy versions over the
wireless communications network.

137. A mobile station for receiving a message
over a wireless communications network, the mobile
station comprising:

a receiver configured to receive a first
5 and a second redundancy versions of the message, wherein
the first and second redundancy versions each includes a
subset of systematic bits and a subset of parity bits;

a rate matcher configured to:

generate a first redundancy version
10 puncturing sequence for a first subset of the parity bits
included in the first redundancy version;

identify the first subset of the
parity bits using the first redundancy version puncturing
sequence;

15 generate a second redundancy version
puncturing sequence for a second subset of the parity
bits included in the at least one additional redundancy
version, wherein the second redundancy version puncturing
sequence indicates that bit positions of the first subset
20 of parity bits do not correspond to parity bits in the
second subset;

- 144 -

identify the second subset of the parity bits using the at least a second redundancy version puncturing sequence; and

25 a Turbo decoder configured to decode the identified bits using a Turbo code to produce the message.

138. A rate matcher for generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the rate matcher configured to:

5 select a first subset of the parity bits;
 generate a first redundancy version of the message, the first redundancy version including the first subset of the parity bits and a first subset of the systematic bits;

10 select a second subset of the parity bits, wherein the parity bits that are selected for the first subset are excluded from being selected for the second subset; and

 generate a second redundancy version of
15 the message, the second redundancy version including the second subset of the parity bits and a first subset of the systematic bits.

139. A wireless communications network for transmitting and receiving a message, the network comprising:

 a transmitting station comprising:
5 a Turbo encoder configured to encode the message to produce encoded data, the encoded data

- 145 -

comprising a plurality of systematic bits and a plurality of parity bits;

a rate matcher configured to:

```

10             select a first subset of the
            parity bits for transmission over the wireless
            communications network;

```

```

                                generate a first redundancy
version of the message, the first redundancy version
15 including the first subset of the parity bits and a first
subset of the systematic bits;

```

select a second subset of the parity bits for transmission over the wireless communications network, wherein the parity bits that are selected for the first subset are excluded from being selected for the second subset; and

generate a second redundancy
version of the message, the second redundancy version
including the second subset of the parity bits and a
25 first subset of the systematic bits;

a transmitter configured to transmit the first and the second redundancy versions over the wireless communications network;

a receiving station comprising:

30 a receiver configured to receive a
first and a second redundancy versions of the message,
wherein the first and second redundancy versions each
includes a subset of systematic bits and a subset of
parity bits;

```
35         a rate matcher configured to:
```

- 146 -

generate a first redundancy
version puncturing sequence for a first subset of the
parity bits included in the first redundancy version;

identify the first subset of the
40 parity bits using the first redundancy version puncturing
sequence;

generate a second redundancy
version puncturing sequence for a second subset of the
parity bits included in the at least one additional
45 redundancy version, wherein the second redundancy version
puncturing sequence indicates that bit positions of the
first subset of parity bits do not correspond to parity
bits in the second subset;

identify the second subset of
50 the parity bits using the at least a second redundancy
version puncturing sequence; and

a Turbo decoder configured to decode
the identified bits using a Turbo code to produce the
message.

140. A method of transmitting a message over a
wireless communications network, the method comprising:

encoding the message using a Turbo code to
produce encoded data, the encoded data comprising a
5 plurality of systematic bits and a plurality of parity
bits;

selecting a first subset of the systematic
bits for transmission over the wireless communications
network;

10 generating a first decodable redundancy
version of the message that includes the first subset of

- 147 -

the systematic bits and a first subset of the parity bits;

15 selecting a second subset and a third subset of the systematic bits for transmission over the wireless communications network, wherein all of the systematic bits corresponding to the message that are not included in the first subset is selected for at least one of the second and the third subsets;

20 generating a second redundancy version of the message that includes the second subset of systematic bits and a second subset of the parity bits that are different from the first subset of parity bits;

25 generating a third redundancy version of the message that includes the third subset of systematic bits and a third subset of the parity bits, wherein a combination of the second and the third redundancy versions is decodable without the first decodable redundancy version;

30 transmitting at least the first redundancy version of the message.

141. The method of claim 140, wherein selecting the first, second, or third subsets of systematic bits comprises puncturing a portion of the systematic bits corresponding to the message such that the portion of systematic bits is excluded from the first, second, or
5 third subsets.

142. The method of claim 140, further comprising:

- 148 -

choosing bits from the encoded data for inclusion in the second subset based at least partly on
5 an expected decoding performance associated with decoding a combination of the first and the second redundancy versions.

143. The method of claim 140, further comprising:

choosing bits from the encoded data for inclusion in the third subset based at least partly on an
5 expected decoding performance associated with decoding a combination of the first and the third redundancy versions.

144. The method of claim 140, wherein the third redundancy version is the same as the first redundancy version.

145. The method of claim 140, wherein the bits are selected for the first and third redundancy versions of the message using a common puncturing loop and using different puncturing loop parameters for each redundancy
5 version.

146. The method of claim 140, further comprising transmitting the first, second, or third redundancy versions over a plurality of time slots defined by a time division multiple access (TDMA)
5 protocol.

147. The method of claim 146, further comprising interleaving the transmitted redundancy versions into a plurality of bursts, wherein each of the

- 149 -

bursts is transmitted over a different one of the time
5 slots.

148. The method of claim 140, further
comprising:

determining whether to transmit a piggy-
backed ACK/NACK (PAN) field in the first, second, or
5 third redundancy versions.

149. The method of claim 148, wherein selecting
the first, second, or third subset of systematic bits
comprises:

puncturing a first portion of the
5 systematic bits such that the first portion is not
included in the first, second, or third subset; and
in response to determining that a PAN
field should be transmitted in the first, second, or
third redundancy versions, puncturing an additional
10 portion of the systematic bits such that the additional
portion is also not included in the first, second, or
third subset.

150. The method of claim 140, further
comprising:

receiving a first negative acknowledgement
using a hybrid automatic repeat request (HARQ) protocol,
5 wherein the second redundancy version is transmitted in
response to receiving the first negative acknowledgment.

151. The method of claim 150, further
comprising:

- 150 -

receiving a second negative
acknowledgement using the HARQ protocol, wherein the
5 third redundancy version is transmitted in response to
receiving the second negative acknowledgment.

152. A method of receiving a message over a
wireless communications network, the method comprising:
receiving a first redundancy version of
the message, wherein the first redundancy version
5 includes a first subset of systematic bits and a first
subset of parity bits;
generating a first redundancy version
puncturing sequence for the first subset of systematic
bits;
10 identifying the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
decoding the identified first subset of
the systematic bits and the first subset of parity bits
15 to produce a first estimate of the message based on a
Turbo code;
receiving a second and third redundancy
versions of the message, wherein the second and third
redundancy versions include second and third subsets of
20 the systematic bits and second and third subsets of the
parity bits;
generating additional redundancy version
puncturing sequences for the second and third redundancy
versions, wherein the additional redundancy version
25 puncturing sequences indicate that all systematic bits

- 151 -

not identified within the first subset are included in at least one of the second and the third subsets;

identifying the second and third subsets of the systematic bits using the additional puncturing sequences;

identifying the second and the third subset of the parity bits, wherein the second subset of parity bits that are different from the first subset of parity bits;

combining the identified bits at least from the second and third redundancy versions; and

decoding the combined bits using a Turbo code to produce an additional estimate of the message, wherein a combination of the second and the third redundancy versions would be decodable without the first decodable redundancy version.

153. The method of claim 152, wherein the third redundancy version is the same as the first redundancy version.

154. The method of claim 152, wherein the puncturing sequences for the first and third redundancy versions of the message are generated using a common puncturing loop and using different puncturing loop parameters for each redundancy version.

155. The method of claim 152, wherein generating the first or the additional redundancy version puncturing sequences comprises determining which bit positions of the systematic bits are included in the first, second, or third subset of the systematic bits.

- 152 -

156. The method of claim 152, wherein identifying the first, second, or third subset of the systematic bits or the parity bits comprises unpuncturing the first, second, or third subset of the systematic bits
5 or the parity bits.

157. The method of claim 152, further comprising receiving the first, second, or third redundancy versions over a plurality of time slots defined by a time division multiple access (TDMA)
5 protocol.

158. The method of claim 157, further comprising deinterleaving the received redundancy versions from a plurality of bursts, wherein each of the bursts is received over a different one of the time
5 slots.

159. The method of claim 152, further comprising:
determining whether a piggy-backed
ACK/NACK (PAN) field is included in the first, second, or
5 third redundancy versions.

160. The method of claim 159, wherein selecting the first, second, or third subset of systematic bits comprises:

in response to determining that a PAN
5 field is included in the first, second, or third redundancy versions, modifying the associated redundancy version puncturing sequence to indicate that the first,

- 153 -

second, or third subset of systematic bits includes fewer of the systematic bits.

161. The method of claim 152, further comprising:

transmitting a first negative
acknowledgement using a hybrid automatic repeat request
5 (HARQ) protocol, wherein the second redundancy version is
receiving in response to the first negative
acknowledgment.

162. The method of claim 150, further comprising:

transmitting a second negative
acknowledgement using the HARQ protocol, wherein the
5 third redundancy version is received in response to the
second negative acknowledgment.

163. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

5 selecting a first subset of the systematic
bits;

generating a first decodable redundancy
version of the message that includes the first subset of
the systematic bits and a first subset of the parity
10 bits;

selecting a second subset and a third
subset of the systematic bits, wherein all of the
systematic bits corresponding to the message that are not

- 154 -

included in the first subset is selected for at least one
15 of the second and the third subsets;

generating a second redundancy version of
the message that includes the second subset of systematic
bits and a second subset of the parity bits that are
different from the first subset of parity bits; and

20 generating a third redundancy version of
the message that includes the third subset of systematic
bits and a third subset of the parity bits, wherein a
combination of the second and the third redundancy
versions is decodable without the first decodable
25 redundancy version.

164. A base station for transmitting a message
over a wireless communications network, the base station
comprising:

a Turbo encoder configured to encode the
5 message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;

a rate matcher configured to:
select a first subset of the
10 systematic bits for transmission over the wireless
communications network;

generate a first decodable redundancy
version of the message, the first redundancy version
including the first subset of the systematic bits and a
15 first subset of the parity bits;

select a second subset and a third
subset of the systematic bits for transmission over the
wireless communications network, wherein the systematic

- 155 -

bits that are not included in the first subset are
20 selected for at least one of the second and the third
subsets;

generate a second redundancy version
of the message, the second redundancy version including
the second subset of systematic bits and a second subset
25 of the parity bits that are different from the first
subset of parity bits;

generate a third redundancy version
of the message, the third redundancy version including
the third subset of systematic bits and a third subset of
30 the parity bits, wherein a combination of the second and
the third redundancy versions is decodable without the
first decodable redundancy version; and

a transmitter configured to transmit the
first, second, and third redundancy versions over the
35 wireless communications network.

165. A mobile station for receiving a message
over a wireless communications network, the mobile
station comprising:

a receiver configured to receive a first,
5 a second, and a third decodable redundancy version of the
message, wherein the message includes a plurality of
systematic bits and a plurality of parity bits;

a rate matcher configured to:

generate a first redundancy version
10 puncturing sequence for a first subset of systematic bits
included in the first redundancy version;

- 156 -

identify a first subset of the systematic bits using the first redundancy version puncturing sequence;

15 generate additional redundancy version puncturing sequences for the second and third redundancy versions, wherein the additional redundancy version puncturing sequences indicate that all systematic bits not identified within the first subset are included
20 in at least one of the second and third redundancy versions;

 identify, from the second and third redundancy versions of the message, the second and third subsets of the systematic bits using the additional
25 puncturing sequences;

 identify, from the second and third redundancy versions of the message, a second and a third subset of the parity bits, wherein the second subset of parity bits that are different from the first subset of
30 parity bits;

 a Turbo decoder configured to decode the identified bits to produce the message, wherein a combination of the second and the third redundancy versions is decodable without the first decodable
35 redundancy version.

166. A rate matcher for generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the rate matcher configured to:

5 select a first subset of the systematic bits;

- 157 -

generate a first decodable redundancy
version of the message, the first redundancy version
including the first subset of the systematic bits and a
10 first subset of the parity bits;

select a second subset and a third
subset of the systematic bits, wherein the systematic
bits that are not included in the first subset are
selected for at least one of the second and the third
15 subsets;

generate a second redundancy version
of the message, the second redundancy version including
the second subset of systematic bits and a second subset
of the parity bits that are different from the first
20 subset of parity bits; and

generate a third redundancy version
of the message, the third redundancy version including
the third subset of systematic bits and a third subset of
the parity bits, wherein a combination of the second and
25 the third redundancy versions is decodable without the
first decodable redundancy version.

167. A wireless communications network for
transmitting and receiving a message, the network
comprising:

a transmitting station comprising:
5 a Turbo encoder configured to encode
the message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;

a rate matcher configured to:

- 158 -

10 select a first subset of the
systematic bits for transmission over the wireless
communications network;
 generate a first decodable
redundancy version of the message, the first redundancy
15 version including the first subset of the systematic bits
and a first subset of the parity bits;
 select a second subset and a
third subset of the systematic bits for transmission over
the wireless communications network, wherein the
20 systematic bits that are not included in the first subset
are selected for at least one of the second and the third
subsets;
 generate a second redundancy
version of the message, the second redundancy version
25 including the second subset of systematic bits and a
second subset of the parity bits that are different from
the first subset of parity bits;
 generate a third redundancy
version of the message, the third redundancy version
30 including the third subset of systematic bits and a third
subset of the parity bits, wherein a combination of the
second and the third redundancy versions is decodable
without the first decodable redundancy version;
 a transmitter configured to transmit
35 the first, second, and third redundancy versions over the
wireless communications network;
 a receiving station comprising:
 a receiver configured to receive a
first, a second, and a third decodable redundancy version

- 159 -

40 of the message, wherein the message includes a plurality
of systematic bits and a plurality of parity bits;
a rate matcher configured to:
generate a first redundancy
version puncturing sequence for a first subset of
45 systematic bits included in the first redundancy version;
identify a first subset of the
systematic bits using the first redundancy version
puncturing sequence;
generate additional redundancy
50 version puncturing sequences for the second and third
redundancy versions, wherein the additional redundancy
version puncturing sequences indicate that all systematic
bits not identified within the first subset are included
in at least one of the second and third redundancy
55 versions;
identify, from the second and
third redundancy versions of the message, the second and
third subsets of the systematic bits using the additional
puncturing sequences;
60 identify, from the second and
third redundancy versions of the message, a second and a
third subset of the parity bits, wherein the second
subset of parity bits that are different from the first
subset of parity bits;
65 a Turbo decoder configured to decode
the identified bits to produce the message, wherein a
combination of the second and the third redundancy
versions is decodable without the first decodable
redundancy version.

- 160 -

168. A method of transmitting a message over a wireless communications network, the method comprising:

encoding the message using a Turbo code to produce encoded data, the encoded data comprising a plurality of systematic bits and a plurality of parity bits;

selecting a first subset of the systematic bits for transmission over the wireless communications network;

generating a first decodable redundancy version of the message that includes the first subset of the systematic bits and a first subset of the parity bits, wherein the first subset of the systematic bits includes less than all of the plurality of systematic bits;

selecting, for transmission over the wireless network, a second subset of the systematic bits that has all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

generating a second redundancy version of the message that includes the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of parity bits;

generating a third redundancy version of the message from the first subset of the systematic bits; transmitting at least the first redundancy version.

- 161 -

169. The method of claim 168, the method further comprising transmitting the first redundancy version, the second redundancy version, and the third redundancy version.

170. The method of claim 169, wherein transmitting the third redundancy version comprises retransmitting the first redundancy version.

171. The method of claim 168, the method further comprising:

initializing a swap value that defines a proportion of the systematic bits included in the first
5 subset of the systematic bits; and
selecting the first subset of the systematic bits based on the swap value.

172. The method of claim 168, wherein selecting the first or the second subsets of the systematic bits comprises puncturing a portion of the systematic bits such that the portion is excluded from the first or the
5 second subsets of the systematic bits.

173. The method of claim 168, wherein selecting the second subset of systematic bits comprises:

for each of the systematic bits:
determining whether the systematic bit has
5 been included in the first redundancy version; and
including the systematic bit in the second subset of the systematic bits when the systematic bit has not been included in the first redundancy version.

- 162 -

174. The method of claim 173, wherein selecting the second subset of the systematic bit further comprises:

selectively puncturing the systematic bit
5 when the systematic bit has been included in the first redundancy version.

175. The method of claim 168, further comprising transmitting the first, second, or third redundancy versions over a plurality of time slots defined by a time division multiple access (TDMA)
5 protocol.

176. The method of claim 175, further comprising interleaving the transmitted redundancy versions into a plurality of bursts, wherein each of the bursts is transmitted over a different one of the time
5 slots.

177. The method of claim 168, further comprising:

determining whether to transmit a piggy-backed ACK/NACK (PAN) field in the first, second, or
5 third redundancy versions.

178. The method of claim 177, wherein selecting the first or the second subset of systematic bits comprises:

puncturing a first portion of the
5 systematic bits such that the first portion is excluded from the first subset or the second subset of the systematic bits; and

- 163 -

in response to determining that a PAN
field should be transmitted in the first or the second
10 redundancy versions, puncturing an additional portion of
the systematic bits such that the additional portion is
also excluded from the first subset or the second subset
of the systematic bits.

179. The method of claim 168, further
comprising:

receiving a first negative acknowledgement
using a hybrid automatic repeat request (HARQ) protocol,
5 wherein the second redundancy version is transmitted in
response to receiving the first negative acknowledgment.

180. The method of claim 179, further
comprising:

receiving a second negative
acknowledgement using the HARQ protocol, wherein the
5 third redundancy version is transmitted in response to
receiving the second negative acknowledgment.

181. A method of receiving a message over a
wireless communications network, the method comprising:

receiving a first decodable redundancy
version of the message, wherein the first redundancy
5 version includes a first subset of systematic bits and a
first subset of parity bits;

generating a first redundancy version
puncturing sequence for the first subset of the
systematic bits, wherein the first redundancy version
10 puncturing sequence indicates that the first subset of

- 164 -

systematic bits includes less than all of the plurality of systematic bits;

identifying, from the first decodable redundancy version, the first subset of the systematic bits using the first redundancy version puncturing sequence;

decoding the identified first subset of the systematic bits and the first subset of parity bits to produce a first estimate of the message based on a Turbo code;

receiving a second redundancy version of the message, wherein the second redundancy version includes a second subset of systematic bits and a second subset of the systematic bits;

identifying, from the second redundancy version, a second subset of the systematic bits, wherein the second subset of the systematic bits includes systematic bits corresponding to the message that are not included in the first subset of systematic bits;

combining the first and second subset of systematic bits and the first and second subset of parity bits;

decoding the combined subsets of systematic and parity bits to produce a second estimate of the message based on a Turbo code;

receiving a third redundancy version of the message, wherein the third redundancy version includes the first subset of systematic bits and the first subset of the parity bits;

- 165 -

40 identifying, from the third redundancy
version, the first subset of the systematic bits using
the first redundancy version puncturing sequence;

 combining the identified bits from at
least two of the first, second, and third redundancy
45 versions; and

 decoding the identified bits using a Turbo
code to produce a third estimate of the message.

 182. The method of claim 181, the method
further comprising:

 determining a swap value that defines a
proportion of the systematic bits that are included in
5 the first subset of the systematic bits; and

 identifying the first subset of the
systematic bits based on the swap value.

 183. The method of claim 181, wherein
generating the first redundancy version puncturing
sequence comprises determining which bit positions of the
systematic bits are included in the first subset of the
5 systematic bits.

 184. The method of claim 181, wherein
identifying the first or the second subset of the
systematic bits comprises unpuncturing the first or
second subset of the systematic bits.

 185. The method of claim 181, further
comprising receiving the first, second, or third
redundancy versions over a plurality of time slots

- 166 -

defined by a time division multiple access (TDMA)
5 protocol.

186. The method of claim 185, further
comprising deinterleaving the received redundancy
versions from a plurality of bursts, wherein each of the
bursts is received over a different one of the time
5 slots.

187. The method of claim 181, further
comprising:
determining whether a piggy-backed
ACK/NACK (PAN) field is included in the first, second, or
5 third redundancy versions.

188. The method of claim 187, wherein
identifying the first or the second subset of systematic
bits comprises:
in response to determining that a PAN
5 field should be included in the first or the second
redundancy versions, modifying the associated redundancy
version puncturing sequence to indicate that the first or
the second subset of systematic bits includes fewer of
the systematic bits.

189. The method of claim 181, further
comprising:
transmitting a first negative
acknowledgement using a hybrid automatic repeat request
5 (HARQ) protocol, wherein the second redundancy version is
received in response to the first negative
acknowledgment.

- 167 -

190. The method of claim 181, further comprising:

transmitting a second negative acknowledgement using the HARQ protocol, wherein the
5 third redundancy version is received in response to receiving the second negative acknowledgment.

191. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

5 selecting a first subset of the systematic bits;

generating a first decodable redundancy version of the message that includes the first subset of the systematic bits and a first subset of the parity
10 bits, wherein the first subset of the systematic bits includes less than all of the plurality of systematic bits;

selecting a second subset of the systematic bits that has all of the systematic bits
15 corresponding to the message that are not included in the first subset of the systematic bits;

generating a second redundancy version of the message that includes the second subset of the systematic bits and a second subset of the parity bits
20 that are different from the first subset of parity bits;
and

generating a third redundancy version of the message from the first subset of the systematic bits.

- 168 -

192. A base station for transmitting a message
25 over a wireless communications network, the base station
comprising:

a Turbo encoder configured to encode the
message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
30 of parity bits;

a rate matcher configured to:

select a first subset of the
systematic bits for transmission over the wireless
communications network;

35 generate a first decodable redundancy
version of the message, the first redundancy version
including the first subset of systematic bits and a first
subset of the parity bits, wherein the first subset of
the systematic bit include less than all of the plurality
40 of systematic bits;

select a second subset of the
systematic bits for transmission over the wireless
communications network, the second subset having all of
the systematic bits that are not included in the first
45 redundancy version;

generate a second redundancy version
of the message, the second redundancy version including
the second subset of systematic bits and a second subset
of the parity bits that are different from the first
50 subset of parity bits; and

a transmitter configured to:

- 169 -

transmit the first and the second redundancy versions over the wireless communications network; and

55 retransmit the first redundancy version as a third redundancy version.

193. A mobile station for receiving a message over a wireless communications network, the mobile station comprising:

 a receiver configured to receive a first
5 decodable redundancy version, a second redundancy version, and a third redundancy version of the message, wherein each of the redundancy versions includes a subset of systematic bits and a subset of parity bits;

 a rate matcher configured to:
10 generate a first redundancy version puncturing sequence for a first subset of systematic bits included in the first redundancy version, wherein the first redundancy version puncturing sequence indicates that the first subset of systematic bits includes less
15 than all of the plurality of systematic bits;

 identify, from the first decodable redundancy version, the first subset of the systematic bits using the first redundancy version puncturing sequence;

20 identify, from the second redundancy version, a second subset of the systematic bits, wherein the second subset of the systematic bits includes systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

- 170 -

25 identify, from the third redundancy
version, the first subset of the systematic bits using
the first redundancy version puncturing sequence; and
 a Turbo decoder configured to decode the
identified bits to produce the message.

194. A rate matcher for generating redundancy
versions of a message, wherein the message includes a
plurality of systematic bits and a plurality of parity
bits, the rate matcher configured to:

5 select a first subset of the systematic
bits;

 generate a first decodable redundancy
version of the message, the first redundancy version
including the first subset of systematic bits and a first
10 subset of the parity bits, wherein the first subset of
the systematic bit includes less than all of the
plurality of systematic bits;

 select a second subset of the systematic
bits, the second subset having all of the systematic bits
15 that are not included in the first redundancy version;
and

 generate a second redundancy version of
the message, the second redundancy version including the
second subset of systematic bits and a second subset of
20 the parity bits that are different from the first subset
of parity bits.

195. A wireless communications network for
transmitting and receiving a message, the network
comprising:

- 171 -

a transmitting station comprising:

5 a Turbo encoder configured to encode
the message to produce encoded data, the encoded data
comprising a plurality of systematic bits and a plurality
of parity bits;

a rate matcher configured to:

10 select a first subset of the
systematic bits for transmission over the wireless
communications network;

generate a first decodable
redundancy version of the message, the first redundancy
15 version including the first subset of systematic bits and
a first subset of the parity bits, wherein the first
subset of the systematic bit include less than all of the
plurality of systematic bits;

select a second subset of the
20 systematic bits for transmission over the wireless
communications network, the second subset having all of
the systematic bits that are not included in the first
redundancy version;

generate a second redundancy
25 version of the message, the second redundancy version
including the second subset of systematic bits and a
second subset of the parity bits that are different from
the first subset of parity bits; and

a transmitter configured to:

30 transmit the first and the
second redundancy versions over the wireless
communications network; and

- 172 -

retransmit the first redundancy version as a third redundancy version;

35 a receiving station comprising:

 a receiver configured to receive a first decodable redundancy version, a second redundancy version, and a third redundancy version of the message, wherein each of the redundancy versions includes a subset
40 of systematic bits and a subset of parity bits;

 a rate matcher configured to:
 generate a first redundancy version puncturing sequence for a first subset of systematic bits included in the first redundancy version,
45 wherein the first redundancy version puncturing sequence indicates that the first subset of systematic bits includes less than all of the plurality of systematic bits;

 identify, from the first
50 decodable redundancy version, the first subset of the systematic bits using the first redundancy version puncturing sequence;

 identify, from the second redundancy version, a second subset of the systematic
55 bits, wherein the second subset of the systematic bits includes systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

 identify, from the third
60 redundancy version, the first subset of the systematic bits using the first redundancy version puncturing sequence; and

- 173 -

a Turbo decoder configured to decode the identified bits to produce the message.

196. A method of transmitting a message over a wireless communications network, the method comprising:

encoding the message using a Turbo code to produce encoded data, the encoded data stream comprising
5 a plurality of systematic bits and a plurality of parity bits;

selecting, for transmission over the wireless communications network, a first subset of the systematic bits using a first puncturing algorithm and a
10 first parameterized swap value;

generating a first redundancy version of the message that includes the first subset of the systematic bits and a first subset of the parity bits;

selecting, for transmission over the
15 wireless communications network, a second subset of the systematic bits using a second puncturing algorithm the second subset of systematic bits having all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

20 generating a second redundancy version of the message that includes the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of the parity bits;

25 selecting, for transmission over the wireless communications network, a third subset of the systematic bits using the first puncturing algorithm and a second parameterized swap value, wherein the second

- 174 -

parameterized swap value is different from the first
30 parameterized swap value;
generating a third redundancy version of
the message that includes the third subset of the
systematic bits and a third subset of the parity bits;
and
35 transmitting at least the first redundancy
version of the message.

197. The method of claim 196, wherein selecting
the first, second, or third subsets of systematic bits
comprises puncturing a portion of the systematic bits
such that the portion is excluded from the selected
5 subset.

198. The method of claim 196, wherein selecting
the second subset of systematic bits comprises:
for each of the systematic bits:
determining whether the systematic bit has
5 been included in the first subset of the systematic bits;
and
puncturing the systematic bit when the
systematic bit has been included in the first subset of
the systematic bits.

199. The method of claim 196, wherein selecting
the second subset of parity bits comprises:
for each of the parity bits:
determining whether the parity bit has
5 been included in the first subset of the parity bits; and

- 175 -

excluding the parity bit from the second subset of parity bits when the parity bit has been included in the first subset of the parity bits.

200. The method of claim 196, wherein the second parameterized swap value is greater than the first second parameterized swap value.

201. The method of claim 196, wherein the second parameterized swap value is 0.30, and wherein selecting the third subset of systematic bits comprises excluding 30% of the systematic bits from being included
5 in the third subset.

202. The method of claim 196, further comprising transmitting the first, second, or third redundancy versions of the message over a plurality of time slots defined by a time division multiple access
5 (TDMA) protocol.

203. The method of claim 202, further comprising interleaving the transmitted redundancy versions into a plurality of bursts, wherein each of the bursts is transmitted over a different one of the time
5 slots.

204. The method of claim 196, further comprising:

determining whether to transmit a piggy-backed ACK/NACK (PAN) field in the first, second, or
5 third redundancy versions.

- 176 -

205. The method of claim 204, wherein selecting the first, second, or third subset of systematic bits comprises:

puncturing a first portion of the
5 systematic bits such that the first portion is excluded from the respective subset of systematic bits; and
in response to determining that a PAN field should be transmitted in the first, second, or third redundancy versions, puncturing an additional
10 portion of the systematic bits such that the additional portion is also excluded from the respective subsets of systematic bits.

206. The method of claim 196, further comprising:

receiving a first negative acknowledgement using a hybrid automatic repeat request (HARQ) protocol,
5 wherein the second redundancy version is transmitted in response to receiving the first negative acknowledgment.

207. The method of claim 206, further comprising:

receiving a second negative acknowledgement using the HARQ protocol, wherein the
5 third redundancy version is transmitted in response to receiving the second negative acknowledgment.

208. A method of receiving a message over a wireless communications network, the method comprising:

receiving a first redundancy version of the message, wherein the first redundancy version

- 177 -

5 includes a first subset of systematic bits and a first
subset of parity bits;
generating a first redundancy version
puncturing sequence for the first subset of the
systematic bits using a first puncturing algorithm and a
10 first parameterized swap value;
identifying the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
decoding the identified first subset of
15 the systematic bits and the first subset of the parity
bits to produce a first estimate of the message based on
a Turbo code;
receiving a second redundancy version of
the message, wherein the second redundancy version
20 includes a second subset of the systematic bits and a
second subset of the parity bits;
generating a second redundancy version
puncturing sequence for a second subset of the systematic
bits using a second puncturing algorithm, wherein the
25 second redundancy version puncturing sequence indicates
that the second subset of systematic bits has all of the
systematic bits corresponding to the message that are not
included in the first subset of the systematic bits;
identifying the second subset of the
30 systematic bits using the second redundancy version
puncturing sequence;
combining the first and second subset of
the systematic bits and the first and second subset of
the parity bits;

- 178 -

35 decoding the combined subsets of
systematic and parity bits to produce a second estimate
of the message based on a Turbo code;
 receiving a third redundancy version of
the message, wherein the third redundancy version
40 includes a third subset of systematic bits and a third
subset of the systematic bits;
 generating a third redundancy version
puncturing sequence for a third subset of the systematic
bits using the first puncturing algorithm and a second
45 parameterized swap value, wherein the second
parameterized swap value is different from the first
parameterized swap value;
 identifying the third subset of the
systematic bits using the third redundancy version
50 puncturing sequence;
 combining at least two of the first,
second, and third subsets of the systematic bits and the
first, second, and the third subset of the parity bits;
and
55 decoding the combined subsets of
systematic and parity bits to produce a third estimate of
the message based on a Turbo code.

209. The method of claim 208, wherein the
second parameterized swap value is greater than the first
second parameterized swap value.

210. The method of claim 208, wherein the
second parameterized swap value is 0.30, and wherein
identifying the third subset of systematic bits comprises

- 179 -

determining that 30% of the systematic bits are not
5 included in the third subset.

211. The method of claim 208, wherein
generating the first, second, or third redundancy version
puncturing sequence comprises determining which bit
positions of the systematic bits are included in the
5 first, second, or third subset of the systematic bits.

212. The method of claim 208, wherein
identifying the first, second, or third subset of the
systematic bits comprises unpuncturing the first, second,
or third subset of the systematic bits.

213. The method of claim 208, further
comprising receiving the first, second, or third
redundancy versions of the message over a plurality of
time slots defined by a time division multiple access
5 (TDMA) protocol.

214. The method of claim 213, further
comprising deinterleaving the received redundancy
versions from a plurality of bursts, wherein each of the
bursts is received over a different one of the time
5 slots.

215. The method of claim 208, further
comprising:
determining whether a piggy-backed
ACK/NACK (PAN) field is included in the first, second, or
5 third redundancy versions.

- 180 -

216. The method of claim 215, wherein identifying the first, second, or third subset of systematic bits comprises:

5 in response to determining that a PAN field should be included in the first, second, or third redundancy versions, modifying the associated redundancy version puncturing sequence to indicate that the first, second, or third subset of systematic bits includes fewer of the systematic bits.

217. The method of claim 208, further comprising:

5 transmitting a first negative acknowledgement using a hybrid automatic repeat request (HARQ) protocol, wherein the second redundancy version is received in response to the first negative acknowledgment.

218. The method of claim 208, further comprising:

5 transmitting a second negative acknowledgement using the HARQ protocol, wherein the third redundancy version is received in response to the second negative acknowledgment.

219. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

- 181 -

5 selecting a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;

 generating a first redundancy version of the message that includes the first subset of the
10 systematic bits and a first subset of the parity bits;

 selecting a second subset of the systematic bits using a second puncturing algorithm the second subset of systematic bits having all of the systematic bits corresponding to the message that are not
15 included in the first subset of the systematic bits;

 generating a second redundancy version of the message that includes the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of the parity
20 bits;

 selecting a third subset of the systematic bits using the first puncturing algorithm and a second parameterized swap value, wherein the second parameterized swap value is different from the first
25 parameterized swap value;

 generating a third redundancy version of the message that includes the third subset of the systematic bits and a third subset of the parity bits.

220. A base station for transmitting a message over a wireless communications network, the base station comprising:

 a Turbo encoder configured to encode the
5 message to produce encoded data, the encoded data

- 182 -

comprising a plurality of systematic bits and a plurality of parity bits;

a rate matcher configured to:

select, for transmission over the
10 wireless communications network, a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;
generate a first redundancy version,
the first redundancy version including the first subset
15 of the systematic bits and a first subset of the parity bits;
select, for transmission over the wireless communications network, a second subset of the systematic bits using a second puncturing algorithm, the
20 second subset having all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;
generate a second redundancy version of the message, the second redundancy version including
25 the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of the parity bits;
select, for transmission over the wireless communications network, a third subset of the
30 systematic bits using the first puncturing algorithm and a second parameterized swap value, wherein the second parameterized swap value is different from the first parameterized swap value; and
generate a third redundancy version
35 of the message, the third redundancy version including

- 183 -

the third subset of the systematic bits and a third subset of the parity bits; and

40 a transmitter configured to selectively transmit the first, second, and third redundancy versions over the wireless communications network.

221. A mobile station for receiving a message over a wireless communications network, the mobile station comprising:

5 a receiver configured to receive a first redundancy version, a second redundancy version, and a third redundancy version of the message, wherein the first, second, and third redundancy versions each include a subset of systematic bits and a subset of parity bits;

10 a rate matcher configured to:
generate a first redundancy version puncturing sequence for a first subset of the systematic bits included in the first redundancy version using a first puncturing algorithm and a first parameterized swap value;

15 identify the first subset of the systematic bits using the first redundancy version puncturing sequence;

20 generate a second redundancy version puncturing sequence for a second subset of the systematic bits included in the second redundancy version using a second puncturing algorithm, wherein the second redundancy version puncturing sequence indicates that the second subset of systematic bits has all of the systematic bits corresponding to the message that are not
25 included in the first subset of the systematic bits;

- 184 -

identify the second subset of the systematic bits using the second redundancy version puncturing sequence;

generate a third redundancy version puncturing sequence for a third subset of the systematic bits included in the third redundancy version using the first puncturing algorithm and a second parameterized swap value, wherein the second parameterized swap value is different from the first parameterized swap value; and

identify the third subset of the systematic bits using the third redundancy version puncturing sequence; and

a Turbo decoder configured to decode the identified bits to produce the message.

222. A rate matcher for generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the rate matcher configured to:

select a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;

generate a first redundancy version, the first redundancy version including the first subset of the systematic bits and a first subset of the parity bits;

select a second subset of the systematic bits using a second puncturing algorithm, the second subset having all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

- 185 -

generate a second redundancy version of the message, the second redundancy version including the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of the parity bits;

select a third subset of the systematic bits using the first puncturing algorithm and a second parameterized swap value, wherein the second parameterized swap value is different from the first parameterized swap value; and

generate a third redundancy version of the message, the third redundancy version including the third subset of the systematic bits and a third subset of the parity bits.

223. A wireless communications network for transmitting and receiving a message, the network comprising:

a transmitting station comprising:
a Turbo encoder configured to encode the message to produce encoded data, the encoded data comprising a plurality of systematic bits and a plurality of parity bits;

a rate matcher configured to:
select, for transmission over the wireless communications network, a first subset of the systematic bits using a first puncturing algorithm and a first parameterized swap value;

generate a first redundancy version, the first redundancy version including the first

- 186 -

subset of the systematic bits and a first subset of the parity bits;

select, for transmission over the wireless communications network, a second subset of the systematic bits using a second puncturing algorithm, the second subset having all of the systematic bits corresponding to the message that are not included in the first subset of the systematic bits;

generate a second redundancy version of the message, the second redundancy version including the second subset of the systematic bits and a second subset of the parity bits that are different from the first subset of the parity bits;

select, for transmission over the wireless communications network, a third subset of the systematic bits using the first puncturing algorithm and a second parameterized swap value, wherein the second parameterized swap value is different from the first parameterized swap value; and

generate a third redundancy version of the message, the third redundancy version including the third subset of the systematic bits and a third subset of the parity bits; and

a transmitter configured to selectively transmit the first, second, and third redundancy versions over the wireless communications network;

a receiving station comprising:

a receiver configured to receive a first redundancy version, a second redundancy version,

- 187 -

and a third redundancy version of the message, wherein the first, second, and third redundancy versions each include a subset of systematic bits and a subset of parity bits;

80 a rate matcher configured to:
 generate a first redundancy
version puncturing sequence for a first subset of the
systematic bits included in the first redundancy version
using a first puncturing algorithm and a first
85 parameterized swap value;
 identify the first subset of the
systematic bits using the first redundancy version
puncturing sequence;
 generate a second redundancy
90 version puncturing sequence for a second subset of the
systematic bits included in the second redundancy version
using a second puncturing algorithm, wherein the second
redundancy version puncturing sequence indicates that the
second subset of systematic bits has all of the
95 systematic bits corresponding to the message that are not
included in the first subset of the systematic bits;
 identify the second subset of
the systematic bits using the second redundancy version
puncturing sequence;
100 generate a third redundancy
version puncturing sequence for a third subset of the
systematic bits included in the third redundancy version
using the first puncturing algorithm and a second
parameterized swap value, wherein the second

- 188 -

105 parameterized swap value is different from the first
parameterized swap value; and
identify the third subset of the
systematic bits using the third redundancy version
puncturing sequence; and
110 a Turbo decoder configured to decode
the identified bits to produce the message.

224. A method of transmitting a message over a
wireless communications network, the method comprising:
encoding the message using a Turbo code to
produce encoded data, the encoded data comprising a
5 plurality of systematic bits and a first plurality of
parity bits;
initializing a first set of loop
parameters for puncturing the systematic bits;
using the first set of loop parameters as
10 input, executing a loop algorithm to puncture a subset of
the systematic bits;
initializing a second set of loop
parameters for puncturing the first plurality of parity
bits;
15 using the second set of loop parameters as
input, executing the loop algorithm to puncture a subset
of the first plurality of parity bits; and
combining, for transmission over the
wireless communications network, the systematic bits that
20 have not been punctured and the first plurality of parity
bits that have not been punctured.

- 189 -

225. The method of claim 224, wherein executing the loop algorithm on a plurality of bits comprises:

iteratively looping through the plurality of bits; and

5 for each bit being looped through:
 determining whether to puncture the bit based on a current puncturing ratio; and
 puncturing the bit when the current puncturing ratio is less than a desired puncturing ratio.

226. The method of claim 224, wherein the encoded data further comprises a second plurality of parity bits, the method further comprising:

 initializing a third set of loop
5 parameters for puncturing the second plurality of parity bits;

 using the third set of loop parameters as input, executing the loop algorithm to puncture a subset of the second plurality of bits; and

10 combining the second plurality of parity bits that have not been punctured with the systematic bits that have not been punctured and the first plurality of parity bits that have not been punctured.

227. The method of claim 224, further comprising:

 receiving system-level rate matching parameters, wherein the first and the second set of loop
5 parameters are initialized based on the system-level rate matching parameters.

- 190 -

228. The method of claim 227, wherein
initializing the first or the second set of loop parameters
comprises:

5 computing internal parameters using the
system-level rate matching parameters; and
 calculating the first or the second set of
loop parameters based on the internal parameters.

229. The method of claim 224, wherein the
combined bits are transmitted in a redundancy version of
the message, the method further comprising:

5 determining a type of the redundancy
version; and
 initializing the first and the second set of
loop parameters based on the type of the redundancy
version.

230. The method of claim 224, wherein
initializing the second set of loop parameters comprises:

5 obtaining a result of the loop algorithm
executed on the systematic bits corresponding to the
message; and
 initializing the second set of loop
parameters based on the result.

231. A method of receiving a message over a
wireless communications network, the method comprising:

5 receiving encoded data;
 initializing a first set of loop
parameters;

- 191 -

using the first set of loop parameters as
input, executing a loop algorithm to generate a
systematic bit puncturing sequence;
identifying a plurality of systematic bits
10 within the encoded data based at least in part on the
systematic bit puncturing sequence;
initializing a second set of loop
parameters;
using the second set of loop parameters as
15 input, executing a loop algorithm to generate a parity
bit puncturing sequence; and
identifying a plurality of parity bits
within the encoded data based at least in part on the
parity bit puncturing sequence; and
20 decoding the identified bits using a Turbo
code to produce the message.

232. The method of claim 231, wherein executing
the loop algorithm on a plurality of bits comprises:
iteratively looping through the plurality
of bits; and
5 for each bit being looped through:
determining whether to puncture the
bit based on a current puncturing ratio; and
puncturing the bit when the current
puncturing ratio is less than a desired puncturing ratio.

233. The method of claim 231, further
comprising:
initializing a third set of loop
parameters;

- 192 -

5 using the third set of loop parameters as
input, executing the loop algorithm to generate a second
parity bit puncturing sequence; and
 identifying a second plurality of parity
bits within the encoded data based at least in part on
10 the second parity bit puncturing sequence.

234. The method of claim 231, further
comprising:

 receiving system-level rate matching
parameters, wherein the first and the second set of loop
5 parameters are initialized based on the system-level rate
matching parameters.

235. The method of claim 231, wherein
initialing the first or the second set of loop parameters
comprises:

 computing internal parameters using the
5 system-level rate matching parameters; and
 calculating the first or the second set of
loop parameters based on the internal parameters.

236. The method of claim 231, wherein the
encoded data is received in a redundancy version of the
message, the method further comprising:

 determining a type of the redundancy
5 version; and
 initialing the first and the second set of
loop parameters based on the type of the redundancy
version.

- 193 -

237. The method of claim 231, wherein initializing the second set of loop parameters comprises initializing the second set of loop parameters based on the systematic bit puncturing sequence.

238. A method of generating redundancy versions of a message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the method comprising:

- 5 initializing a first set of loop parameters for puncturing the systematic bits;
 - using the first set of loop parameters as input, executing a loop algorithm to puncture a subset of the systematic bits;
- 10 initializing a second set of loop parameters for puncturing the first plurality of parity bits;
 - using the second set of loop parameters as input, executing the loop algorithm to puncture a subset
- 15 of the first plurality of parity bits; and
 - combining the systematic bits that have not been punctured and the first plurality of parity bits that have not been punctured to generate the redundancy version.

239. A base station for transmitting a message over a wireless communications network, the base station comprising:

- 5 a Turbo encoder configured to encode the message to produce encoded data, the encoded data

- 194 -

comprising a plurality of systematic bits and a first plurality of parity bits;

10 a rate matcher comprising loop logic for puncturing encoded data bits , wherein the rate matcher is configured to:

initialize a first set of loop parameters for puncturing the systematic bits;

15 using the first set of loop parameters as input, execute the loop logic to puncture a subset of the systematic bits;

initialize a second set of loop parameters for puncturing the first plurality of parity bits;

20 using the second set of loop parameters as input, execute the loop logic to puncture a subset of the first plurality of parity bits;

combine the systematic bits that have not been punctured and the first plurality of parity bits that have not been punctured; and

25 a transmitter configured to transmit the combined systematic and parity bits in a redundancy version of the message over the wireless communications network.

240. A mobile station for receiving a message over a wireless communications network, the mobile station comprising:

5 a receiver configured to receive encoded data;

a rate matcher configured to:

- 195 -

```

                                initialize a first set of loop
parameters;
                                execute a loop algorithm to generate
10  a systematic bit puncturing sequence using the first set
of loop parameters as input,;
                                identify a plurality of systematic
bits within the encoded data based at least in part on
the systematic bit puncturing sequence;
15  initialize a second set of loop
parameters;
                                execute a loop algorithm to generate
a parity bit puncturing sequence using the second set of
loop parameters as input;
20  identify a plurality of parity bits
within the encoded data based at least in part on the
parity bit puncturing sequence; and
                                a Turbo decoder configured to decode the
identified bits using a Turbo code to produce the
25  message.
```

- 196 -

241. A rate matcher for generating an encoded message, wherein the message includes a plurality of systematic bits and a plurality of parity bits, the rate matcher configured to:

- 5 initialize a first set of loop parameters for puncturing the systematic bits;
 using the first set of loop parameters as input, execute the loop logic to puncture a subset of the systematic bits;
- 10 initialize a second set of loop parameters for puncturing the first plurality of parity bits;
 using the second set of loop parameters as input, execute the loop logic to puncture a subset of the first plurality of parity bits; and
- 15 combine the systematic bits that have not been punctured and the first plurality of parity bits that have not been punctured.

242. A wireless communications network for transmitting and receiving a message, the network comprising:

- a transmitting station comprising:
 - 5 a Turbo encoder configured to encode the message to produce encoded data, the encoded data comprising a plurality of systematic bits and a first plurality of parity bits;
 - a rate matcher comprising loop logic
 - 10 for puncturing encoded data bits, wherein the rate matcher is configured to:

- 197 -

initialize a first set of loop
parameters for puncturing the systematic bits;
using the first set of loop
15 parameters as input, execute the loop logic to puncture a
subset of the systematic bits;
initialize a second set of loop
parameters for puncturing the first plurality of parity
bits;
20 using the second set of loop
parameters as input, execute the loop logic to puncture a
subset of the first plurality of parity bits;
combine the systematic bits that
have not been punctured and the first plurality of parity
25 bits that have not been punctured;
a transmitter configured to transmit
the combined systematic and parity bits in a redundancy
version of the message over the wireless communications
network;
30 a receiving station comprising:
a receiver configured to receive the
redundancy version;
a rate matcher configured to:
initialize a first set of loop
35 parameters;
execute a loop algorithm to
generate a systematic bit puncturing sequence using the
first set of loop parameters as input,;
identify a plurality of
40 systematic bits within the redundancy version based at
least in part on the systematic bit puncturing sequence;

- 198 -

initialize a second set of loop
parameters;

execute a loop algorithm to
45 generate a parity bit puncturing sequence using the
second set of loop parameters as input;
identify a plurality of parity
bits within the redundancy version based at least in part
on the parity bit puncturing sequence; and
50 a Turbo decoder configured to decode
the identified bits using a Turbo code to produce the
message.

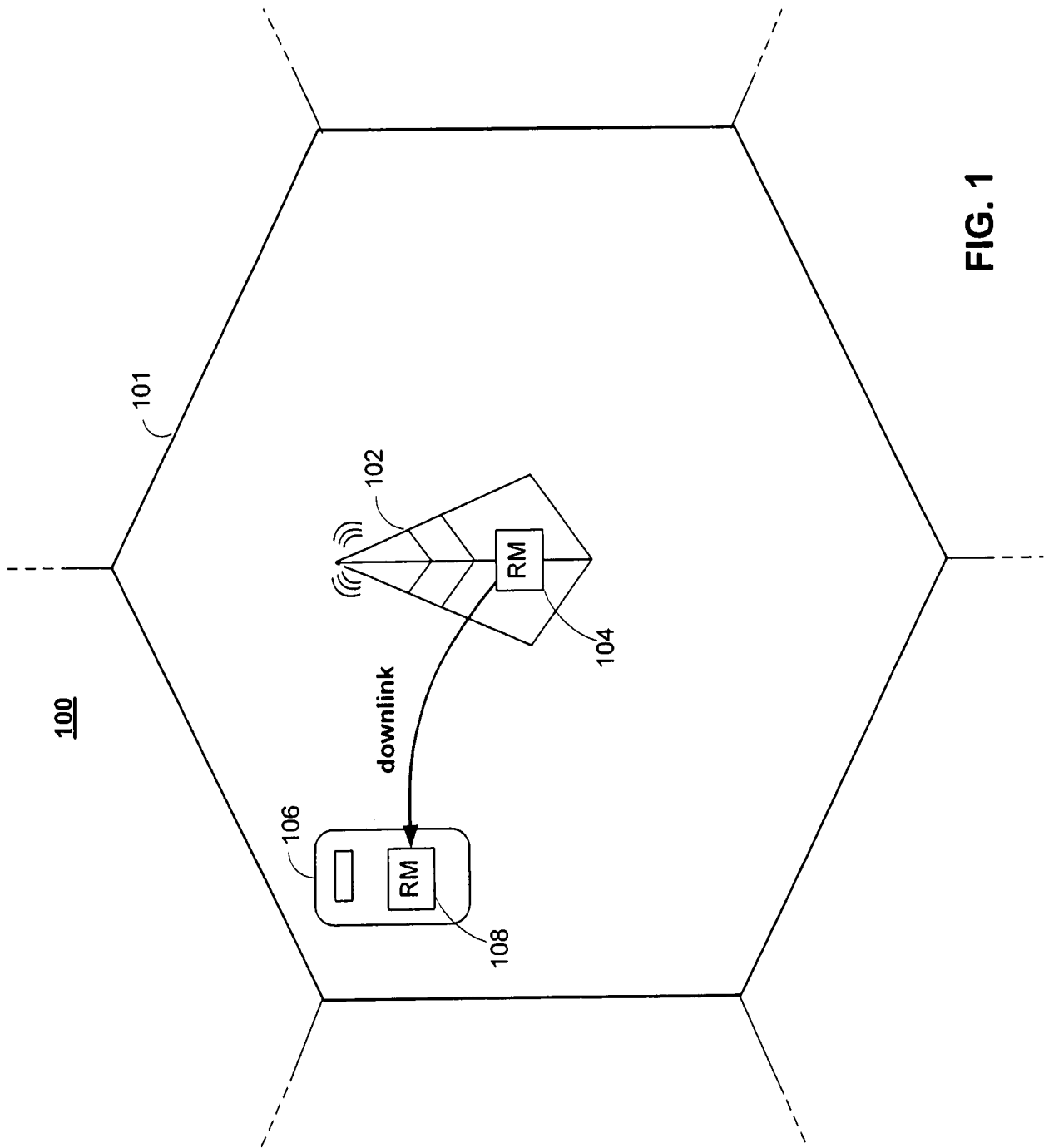


FIG. 1

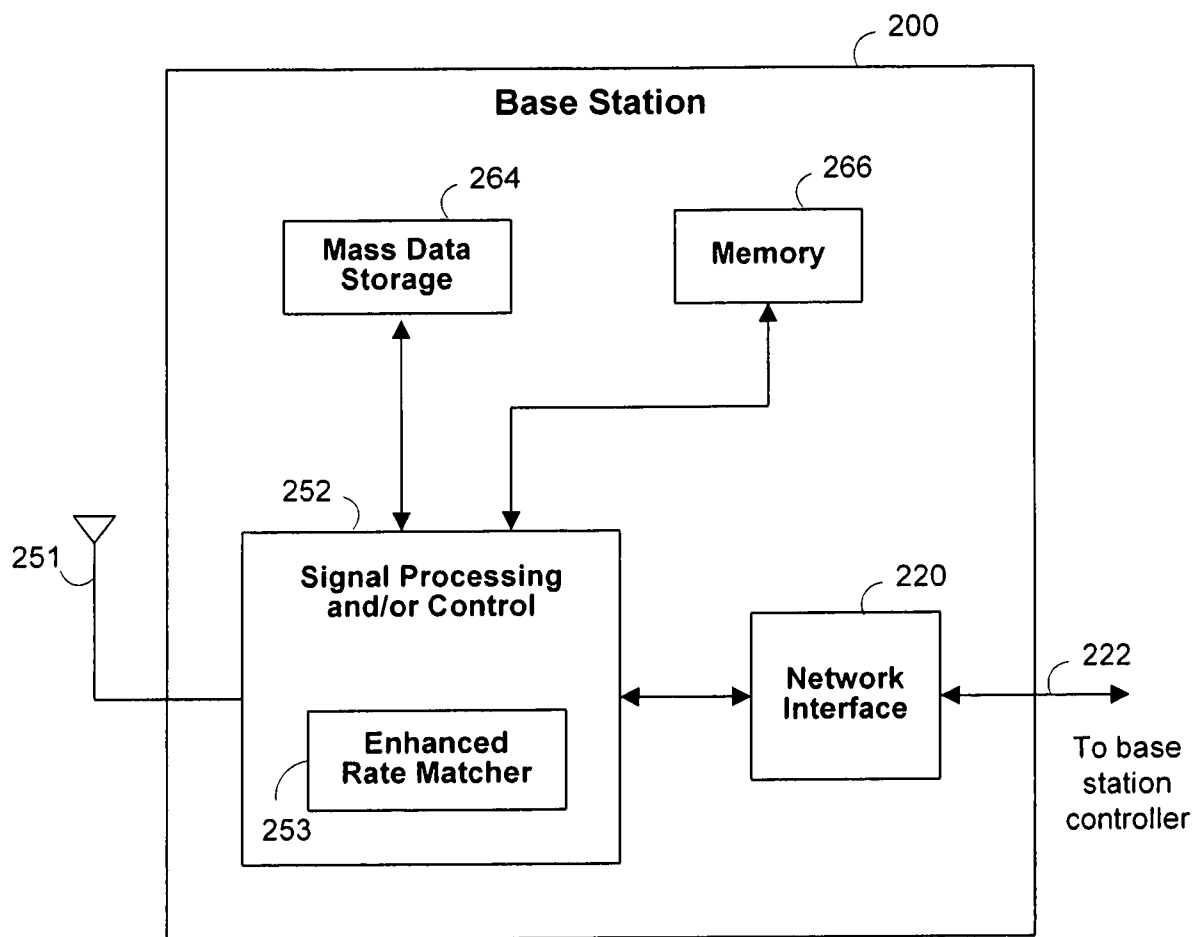


FIG. 2

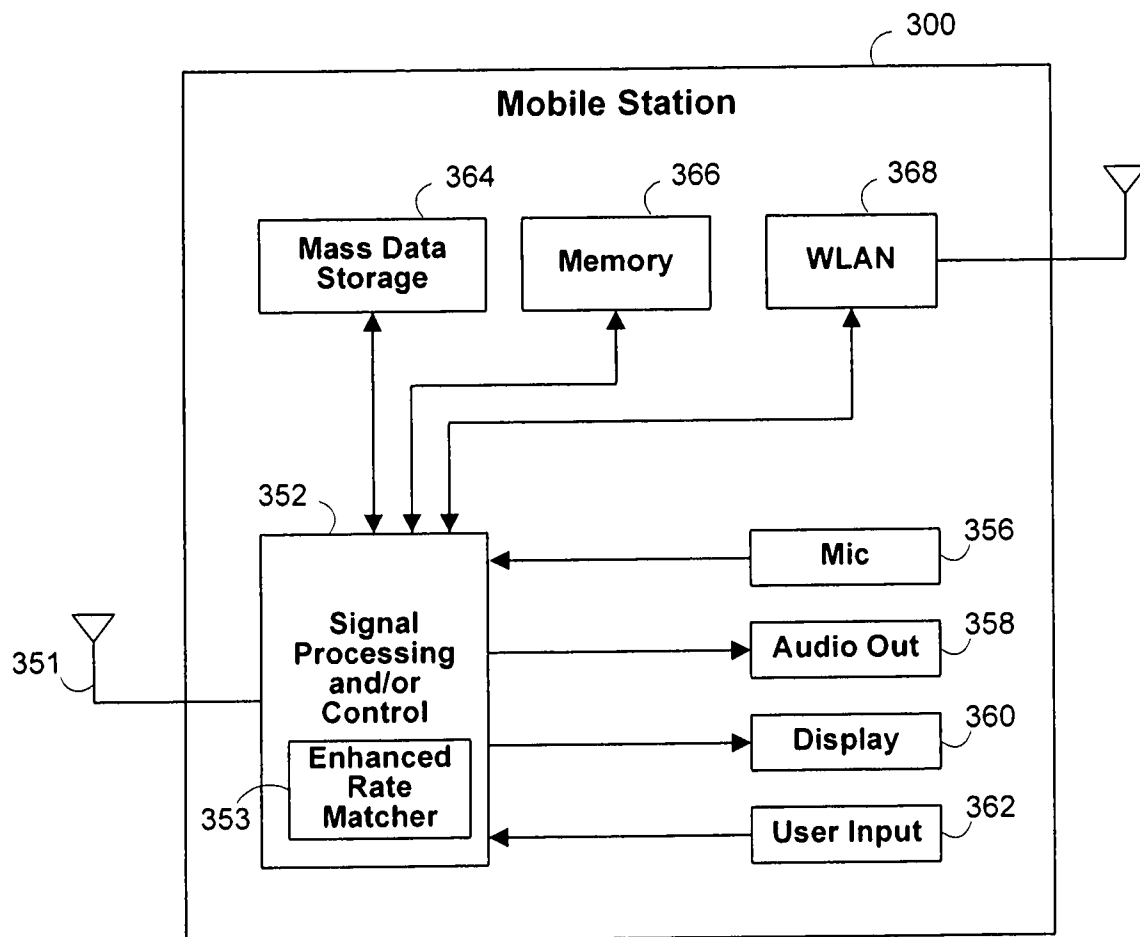


FIG. 3

4/31

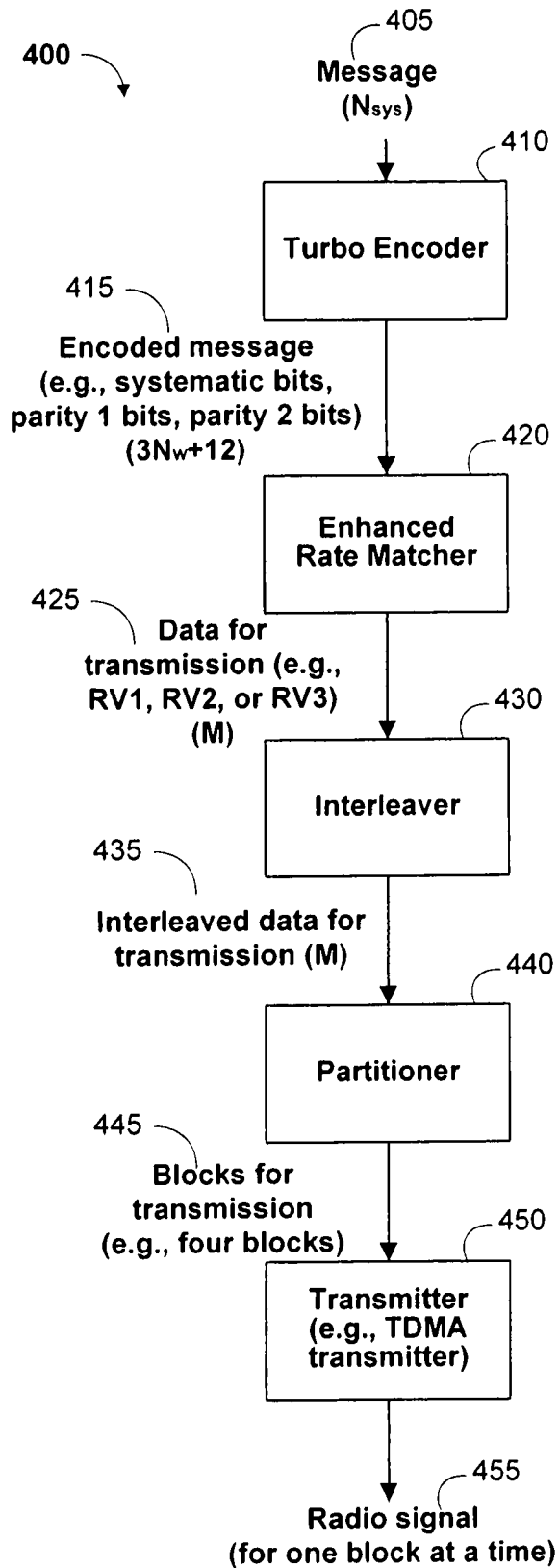


FIG. 4

5/31

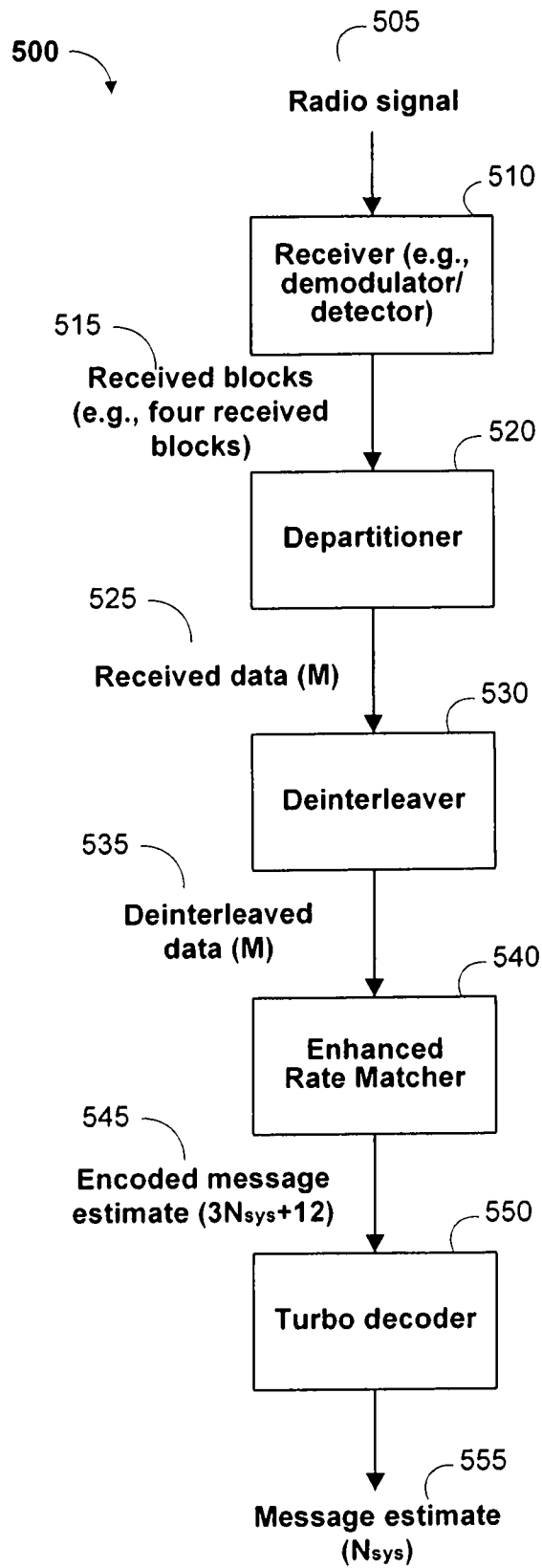
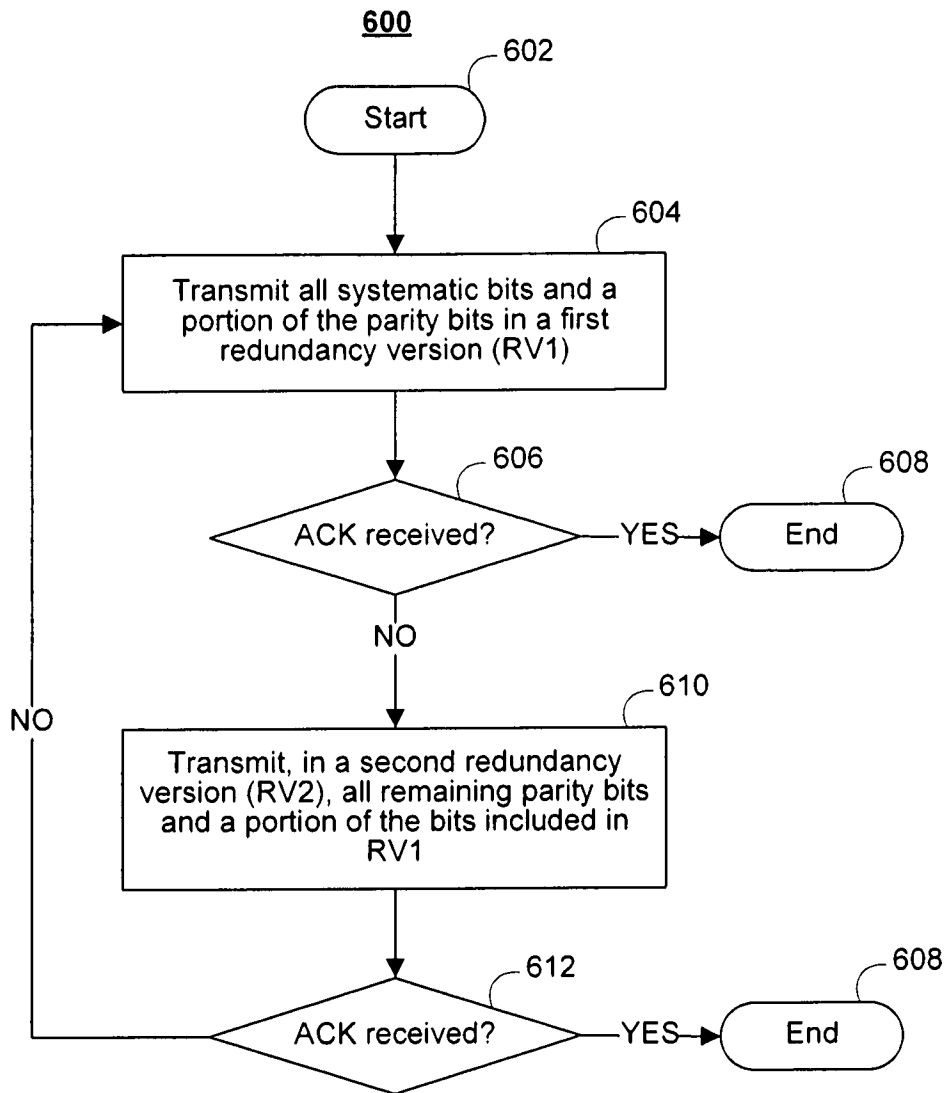
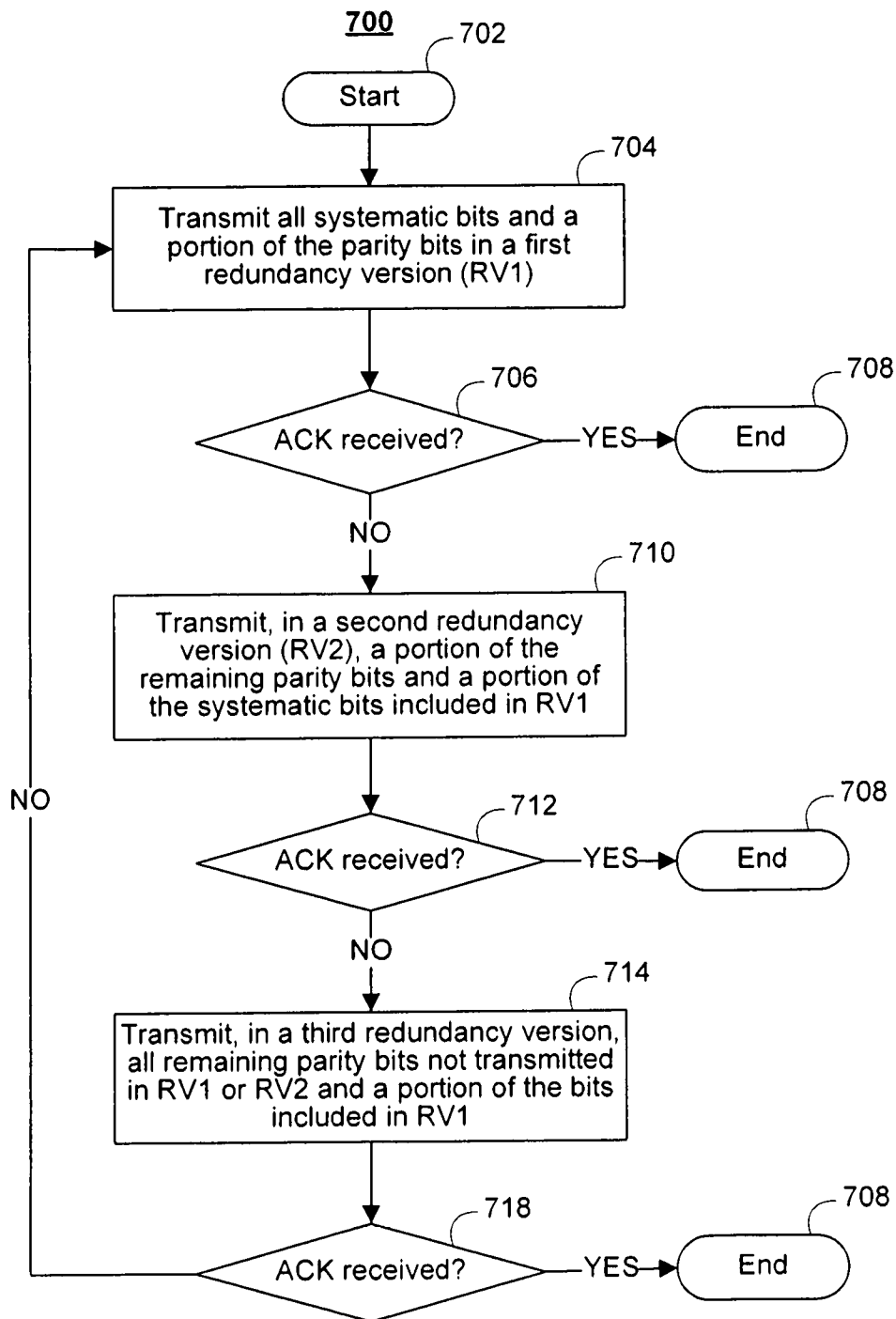


FIG. 5

**FIG. 6**

7/31

**FIG. 7**

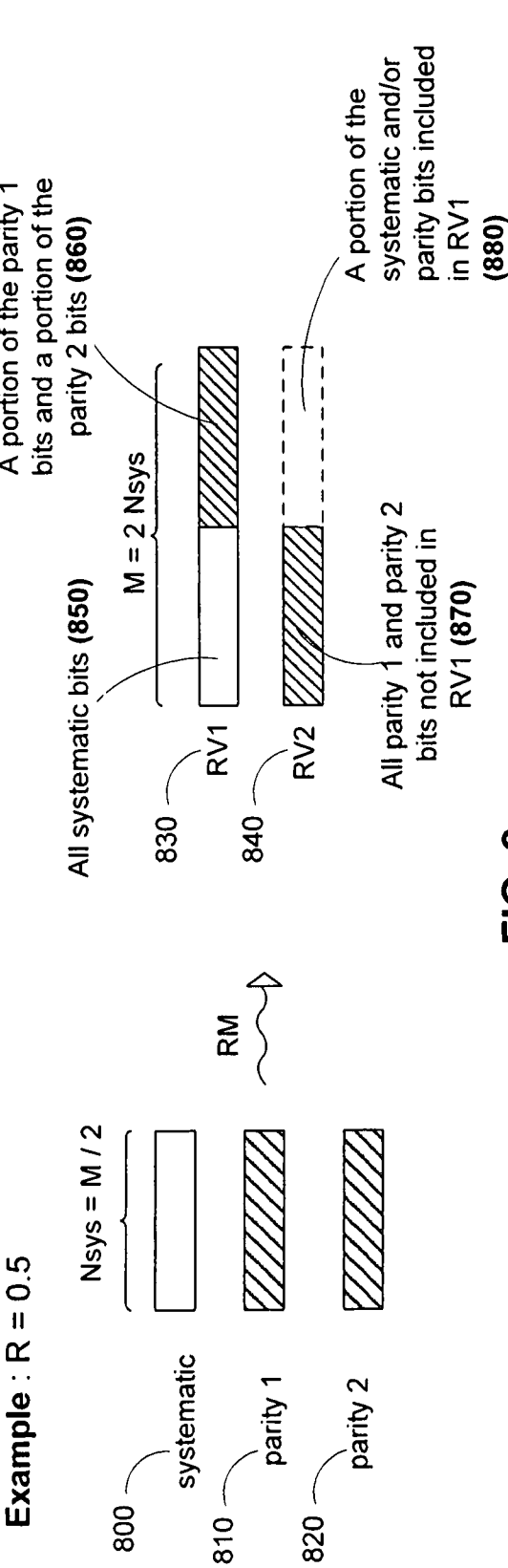


FIG. 8

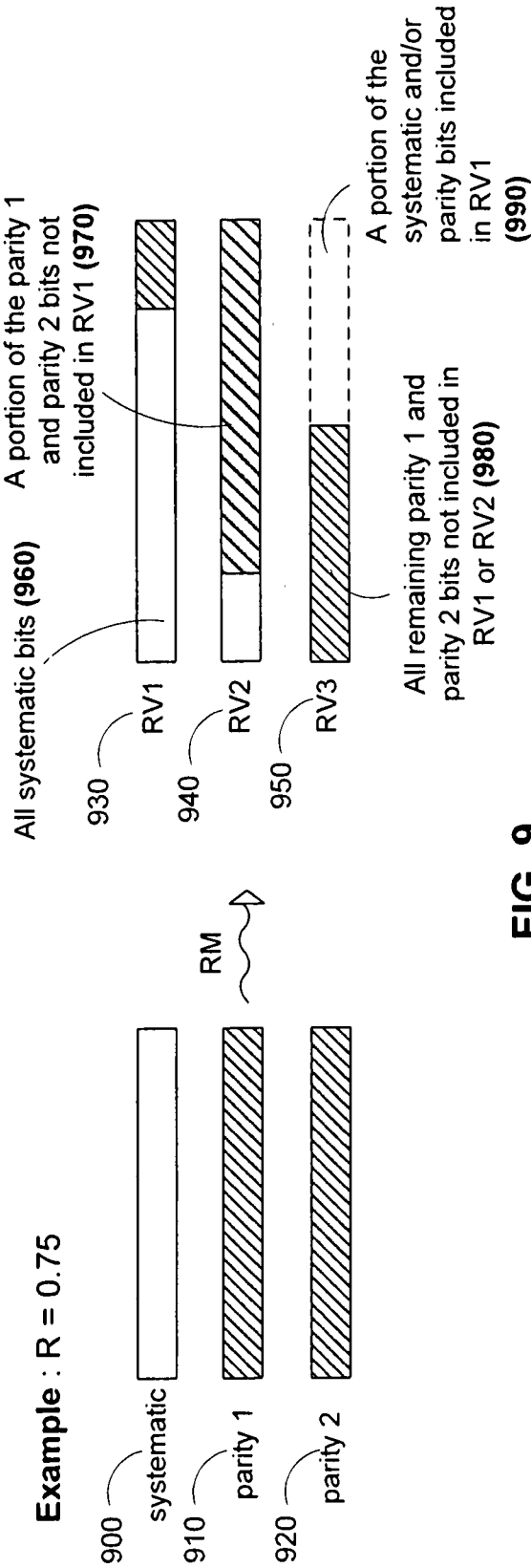
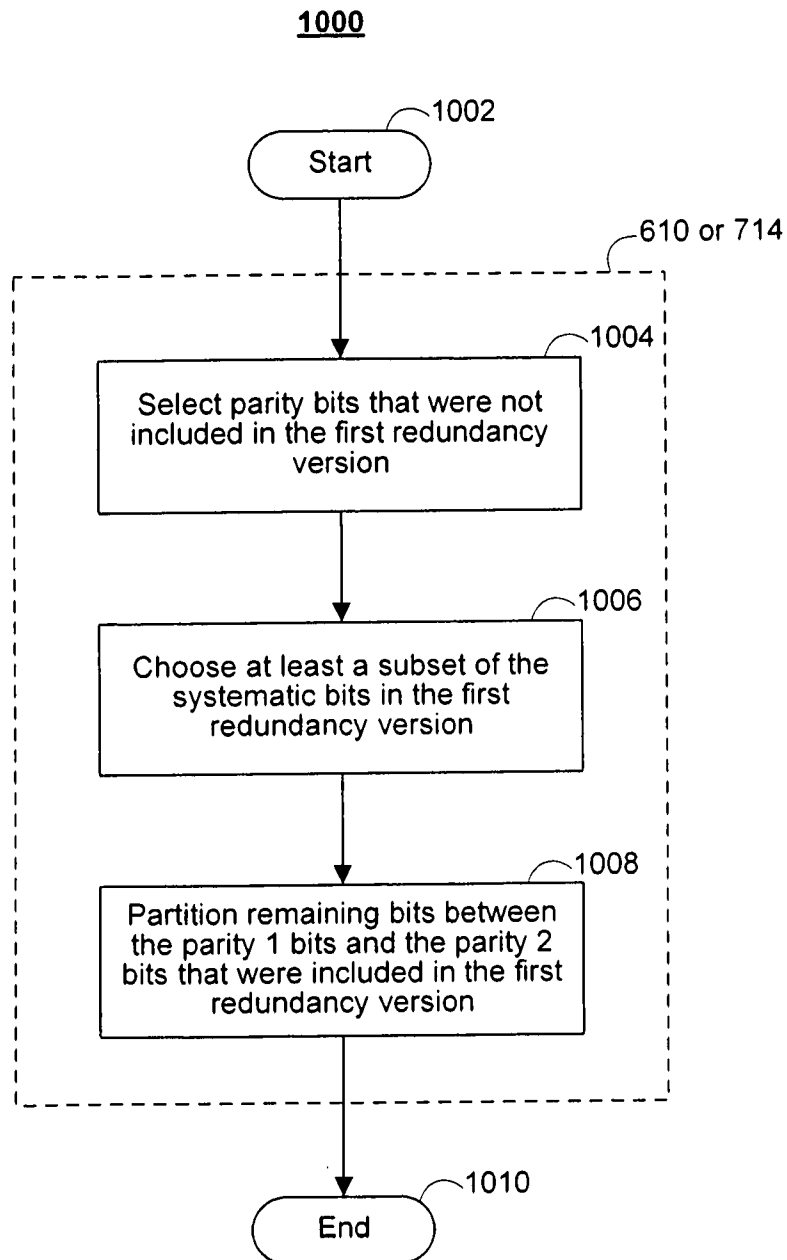
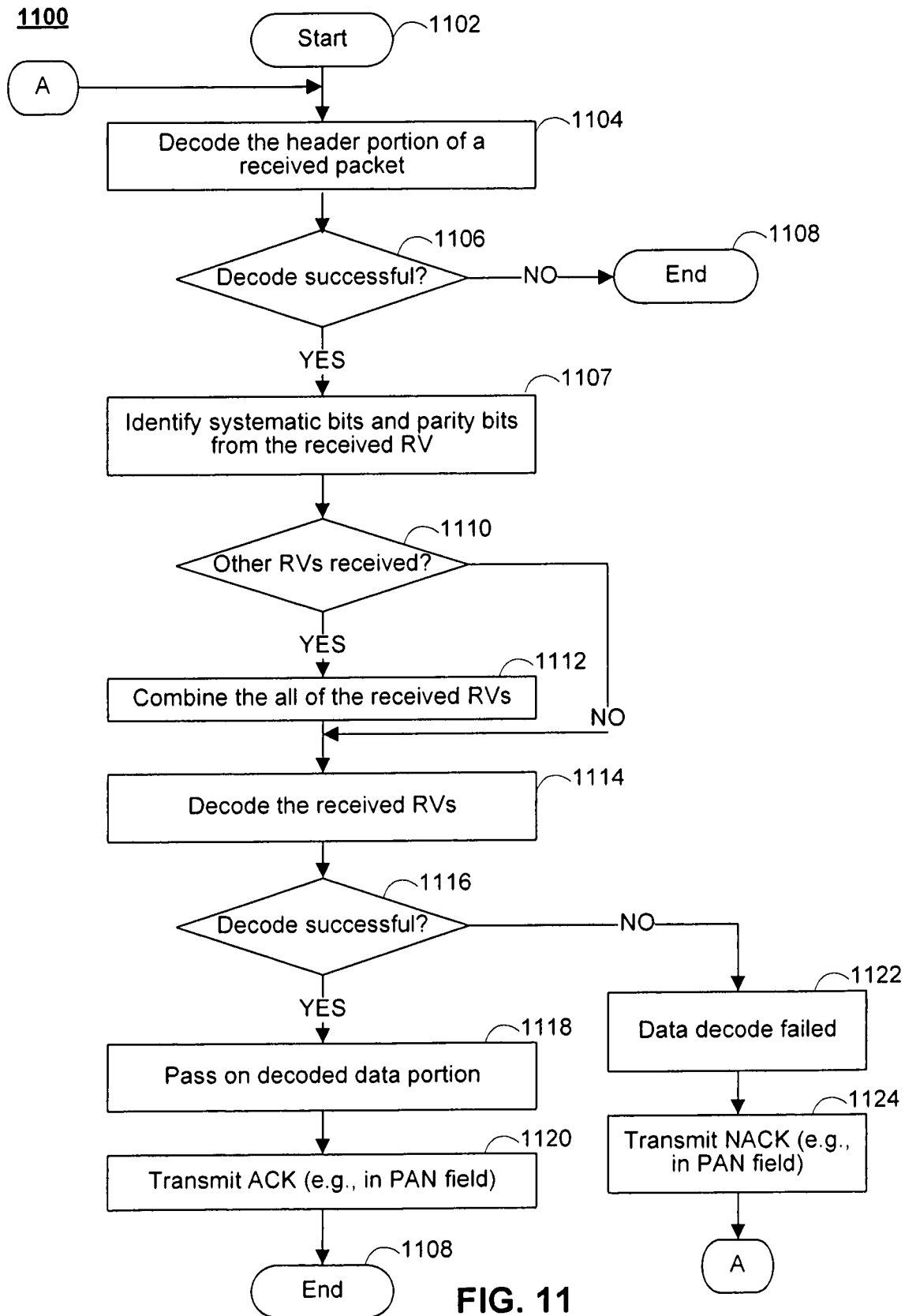


FIG. 9

**FIG. 10**

10/31



11/31

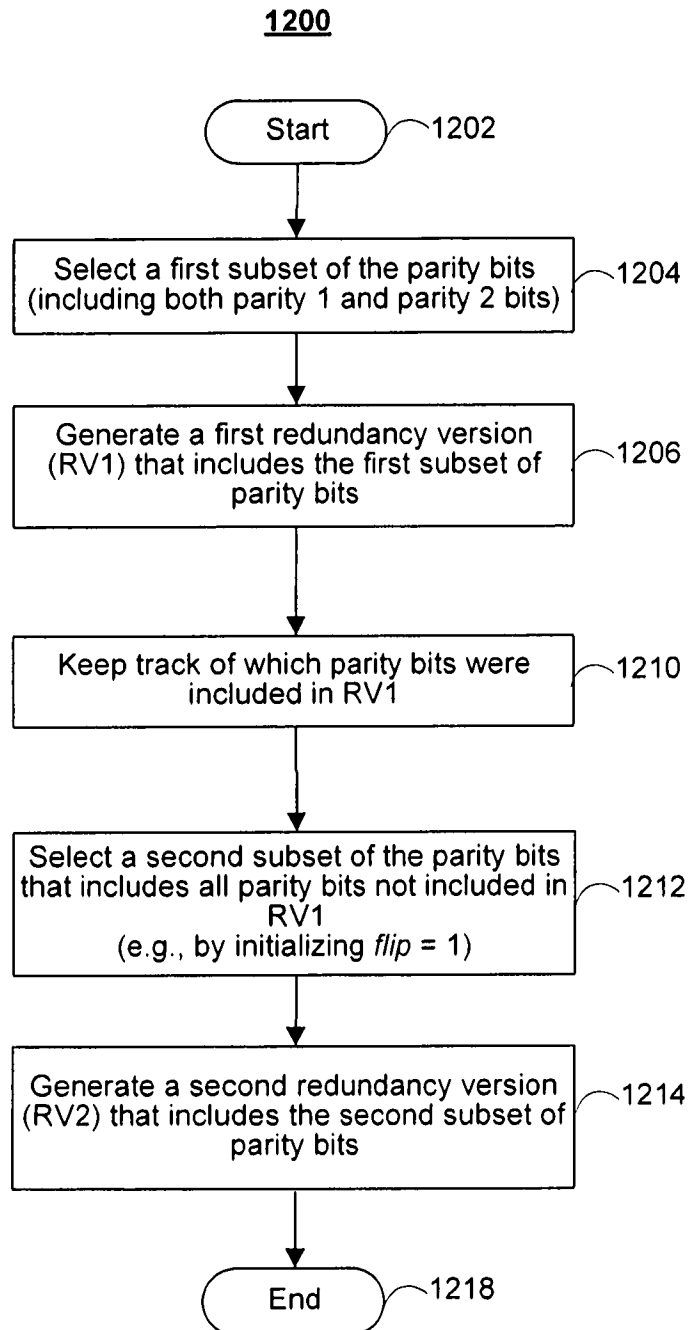


FIG. 12A

12/31

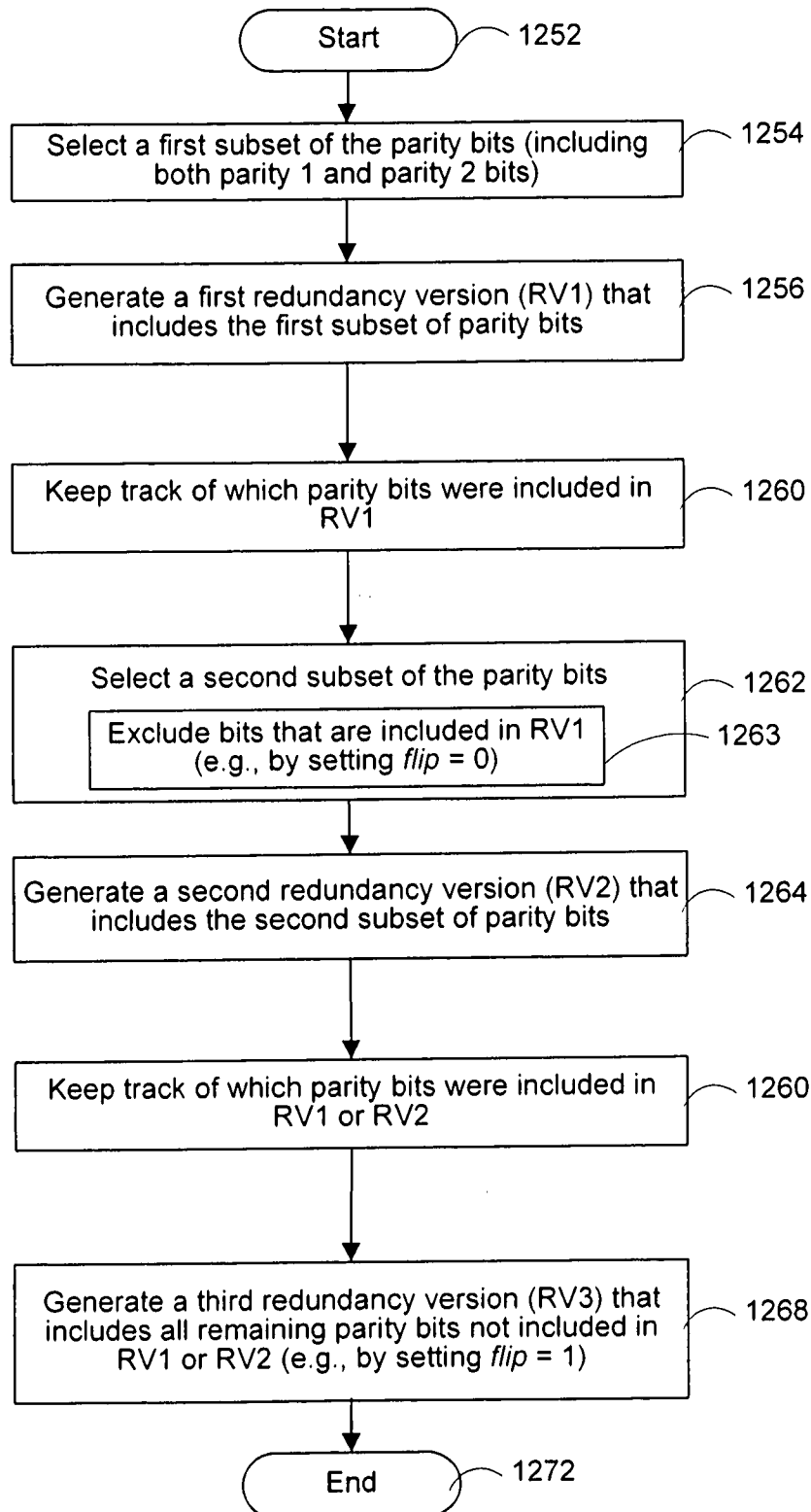
1250

FIG. 12B

Include a bit if the bit was not
previously included

Table 1: *Flip* values

	RV1	RV2 - I	RV2 - II	RV3
Systematic bits	0	1	1	1
Parity 1 bits	0	1	0	1
Parity 2 bits	0	1	0	1

Do not include a bit if the bit
was previously included

Table 2: Rate Matching Parameters

	<i>swap</i>	N _{sys}	M _{NO_PAN}	M _{PAN}
DAS-5	5%	466	1248	1172
DAS-10	15%	674	1060	1022
DBS-5	5%	466	956	876
DBS-6	15%	610	956	876
DBS-8	5%	610	996	956

FIG. 13

14/31

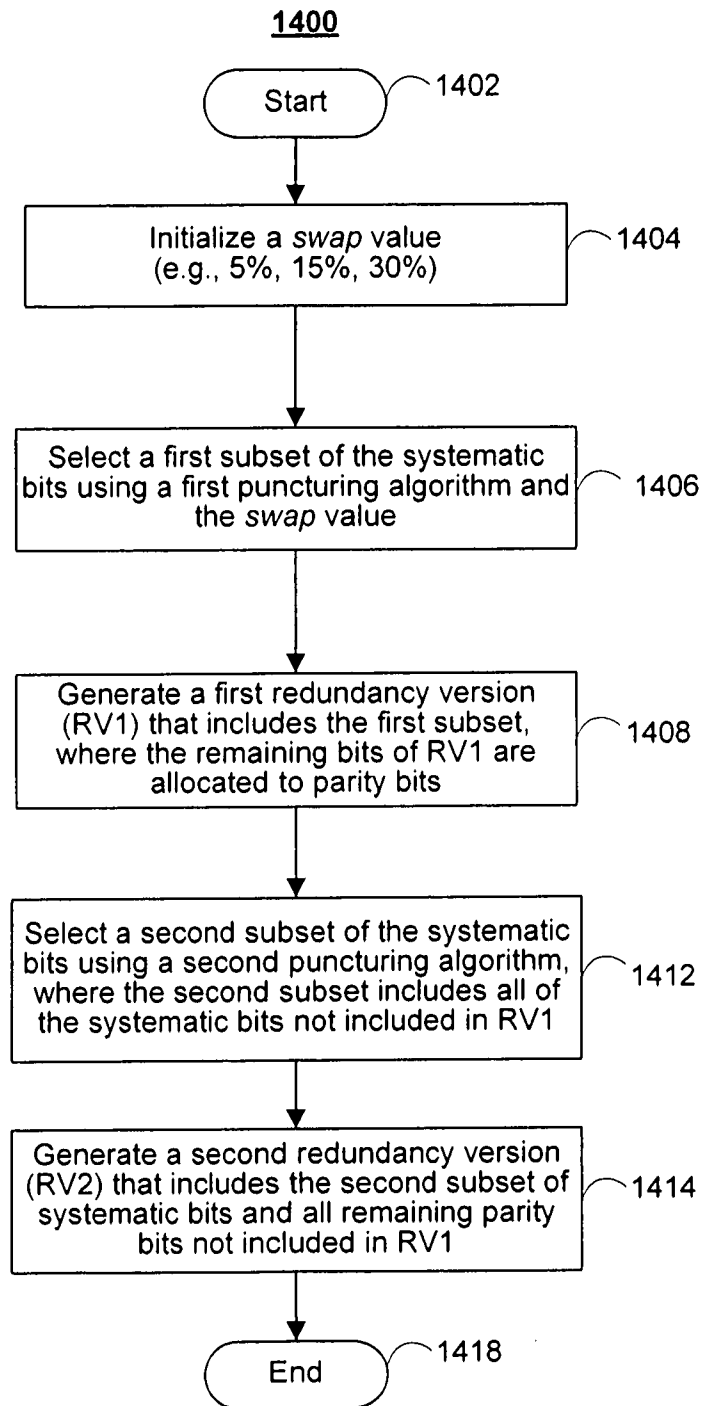


FIG. 14

15/31

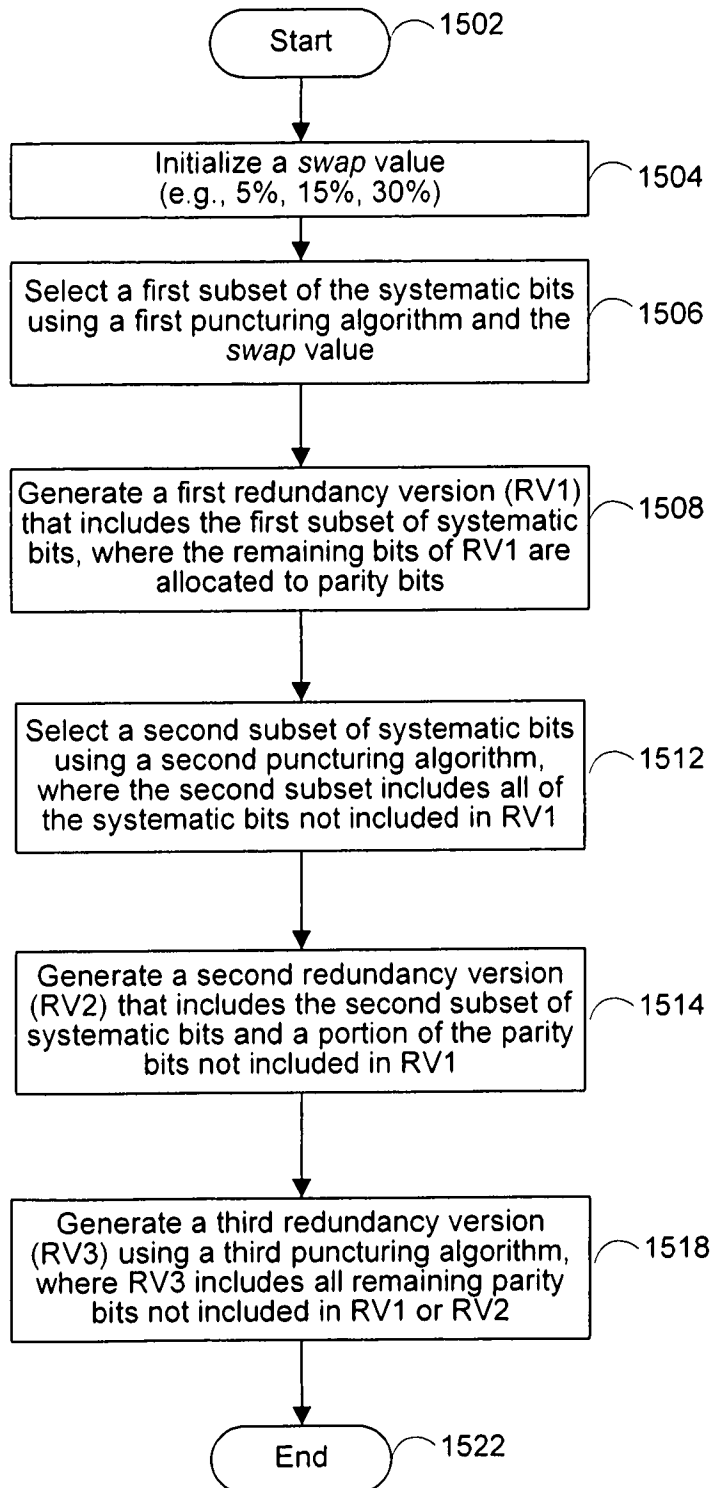
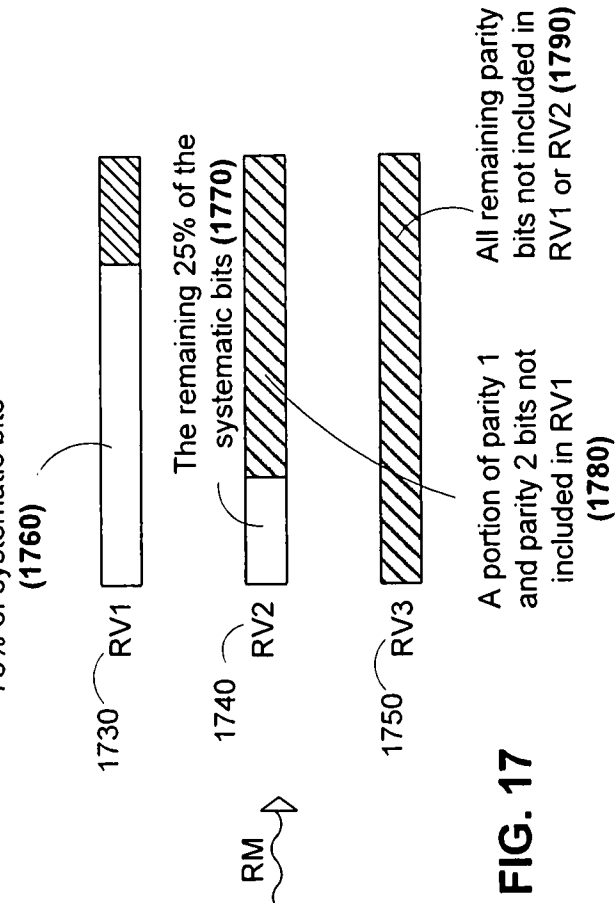
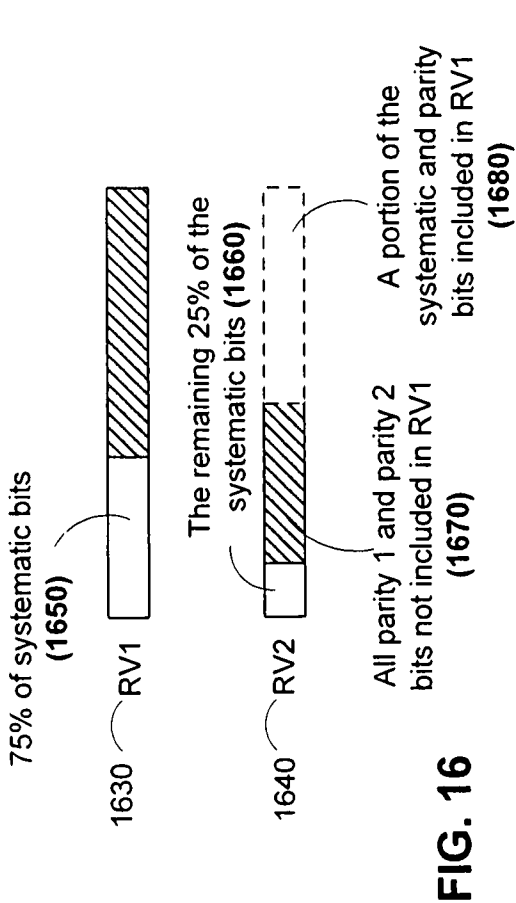
1500

FIG. 15



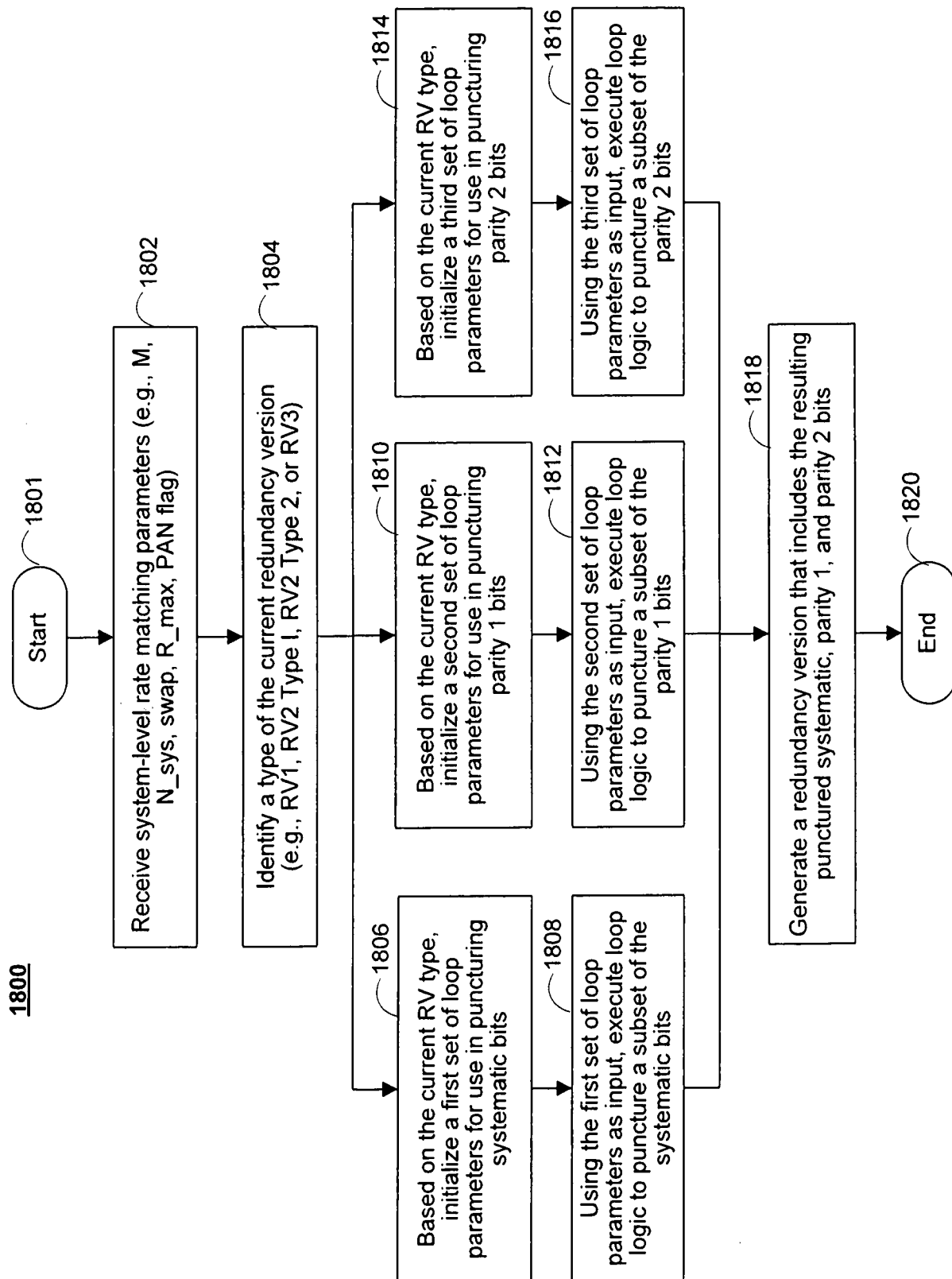


Table 3: Loop parameters

	e_{plus}	e_{minus}	$e2_{plus}$	$e2_{minus}$
Systematic bits	$X_{ir,1}$	$ X_{ir,1} - N_{tr,1} $	$\tilde{X}_{ir,1}$	$ \tilde{X}_{ir,1} - \tilde{N}_{tr,1} $
Parity 1 bits	$2 \cdot X_{ir,2}$	$2 \cdot X_{ir,2} - N_{tr,2} $	$2 \cdot \tilde{X}_{ir,2}$	$2 \cdot \tilde{X}_{ir,2} - \tilde{N}_{tr,2} $
Parity 2 bits	$X_{ir,3}$	$ X_{ir,3} - N_{tr,3} $	$\tilde{X}_{ir,3}$	$ \tilde{X}_{ir,3} - \tilde{N}_{tr,3} $

...
Internal parameters
computed from system-
level rate matching
parameters
...

FIG. 19

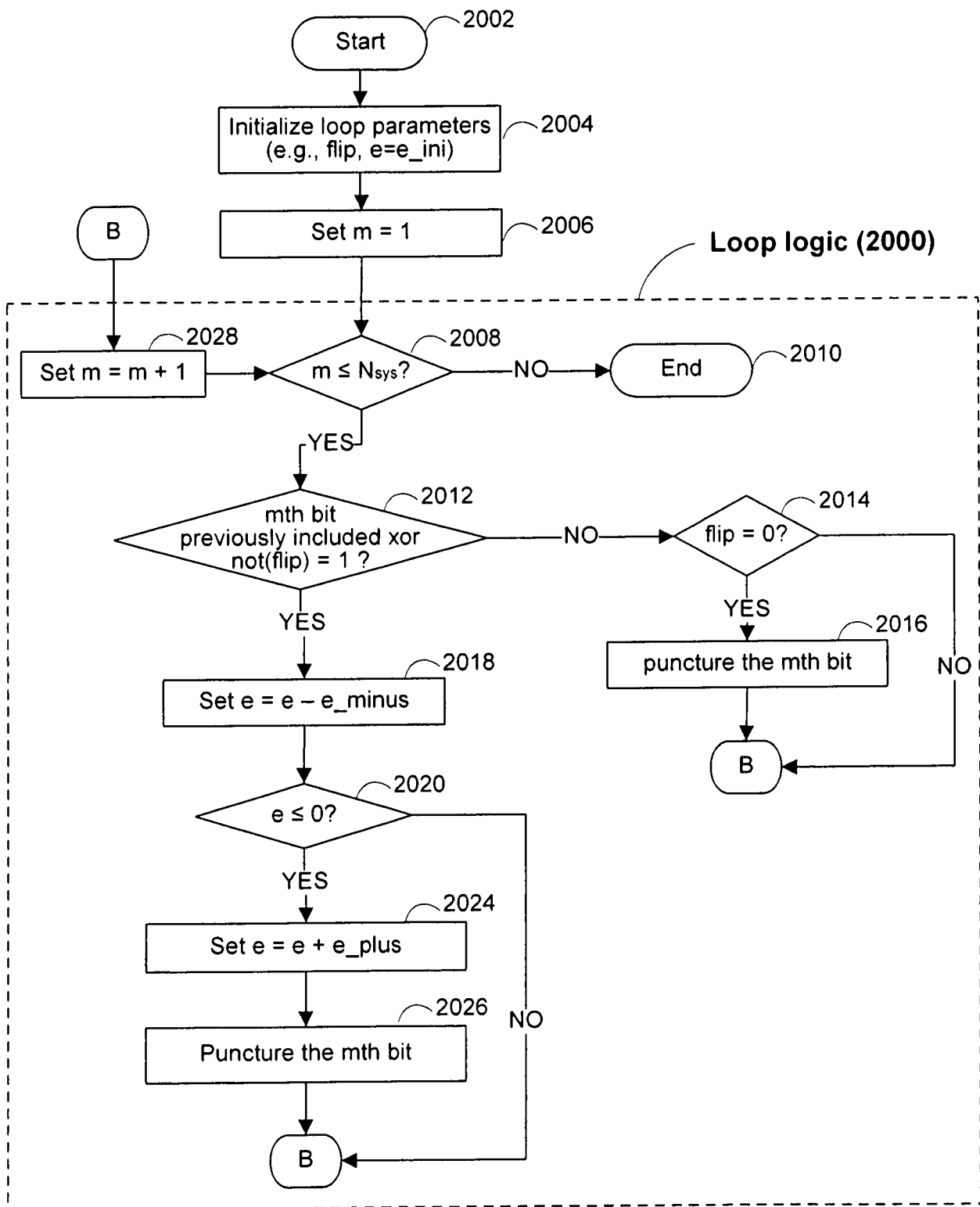


FIG. 20

20/31

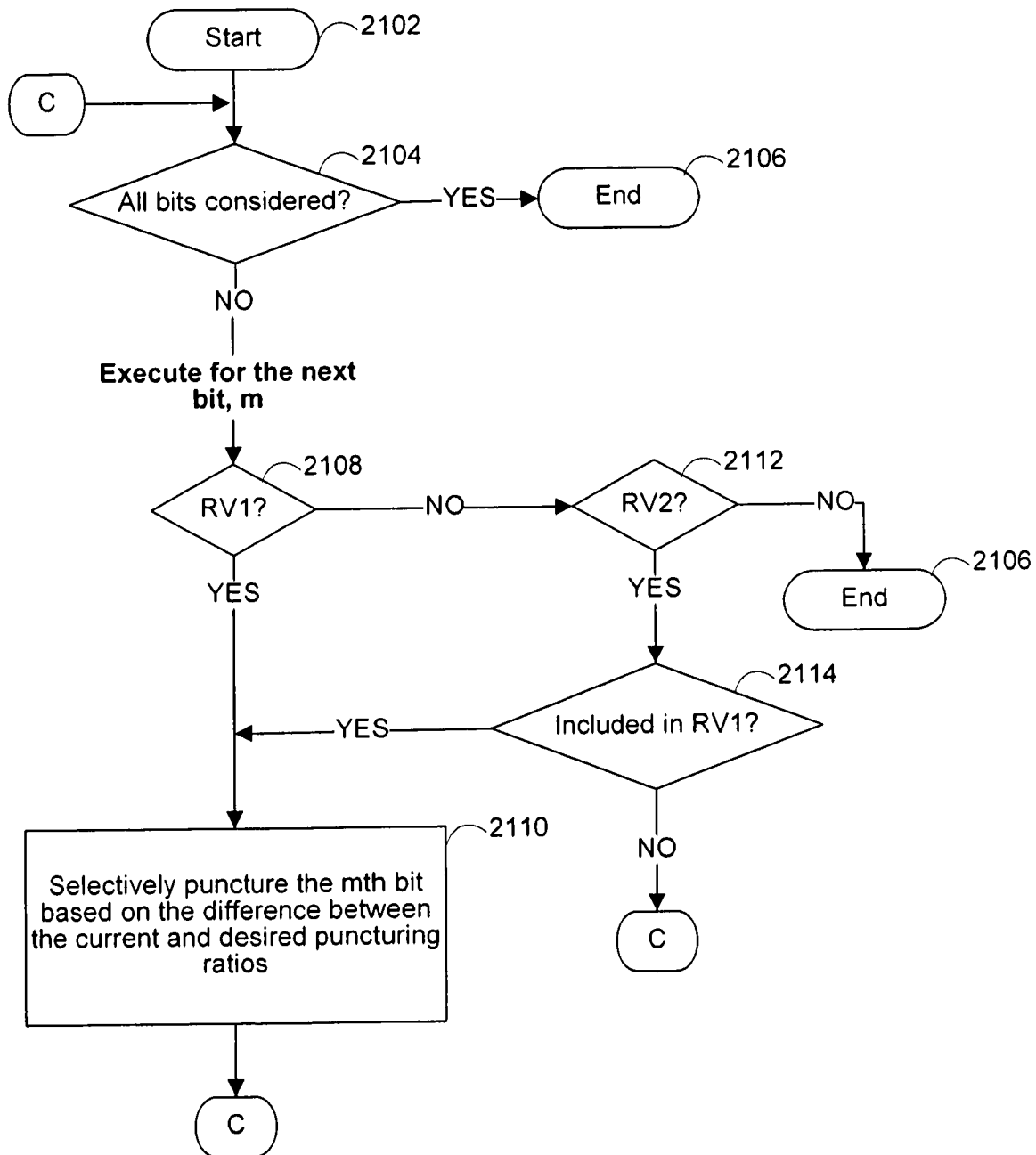
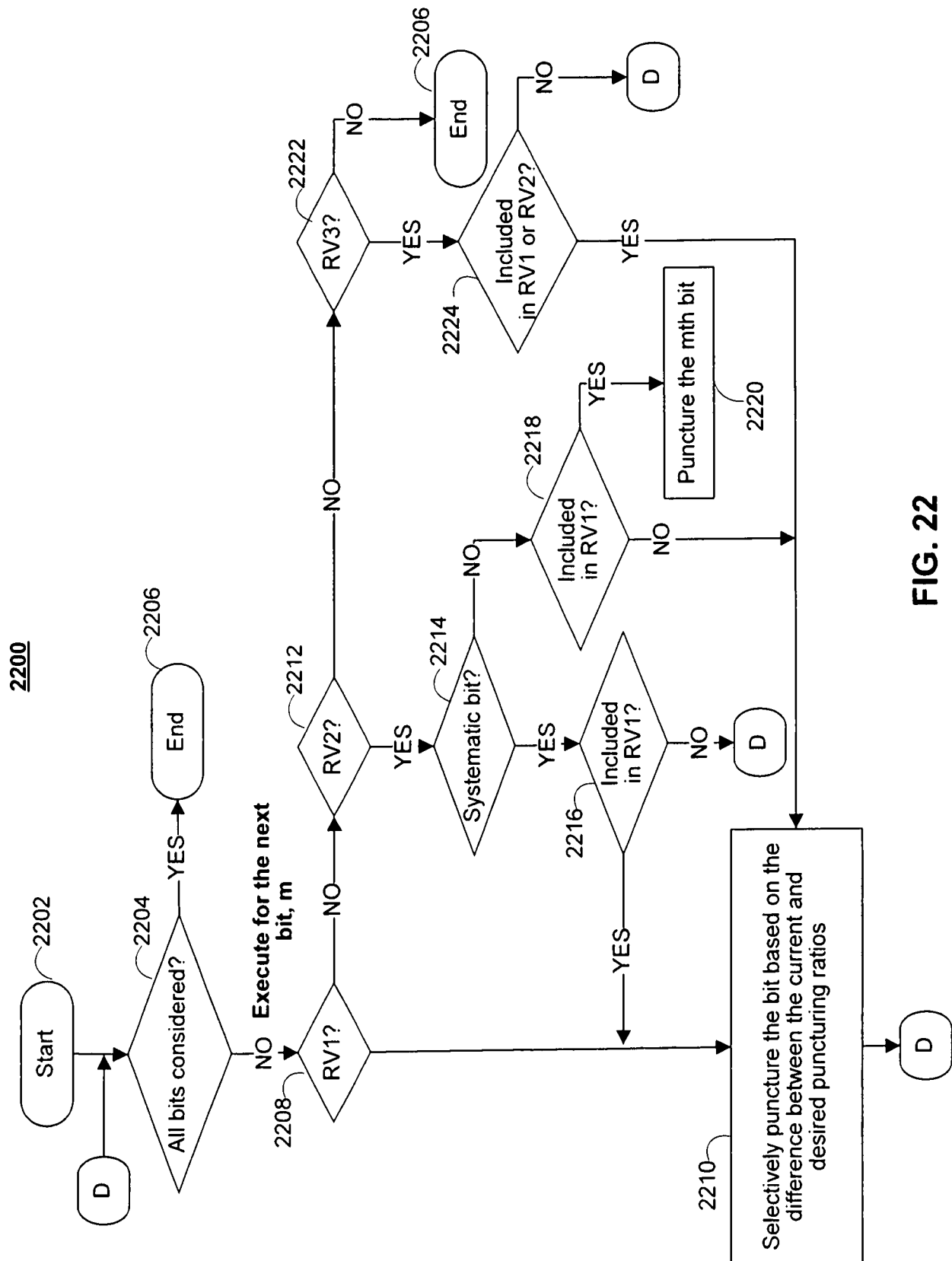
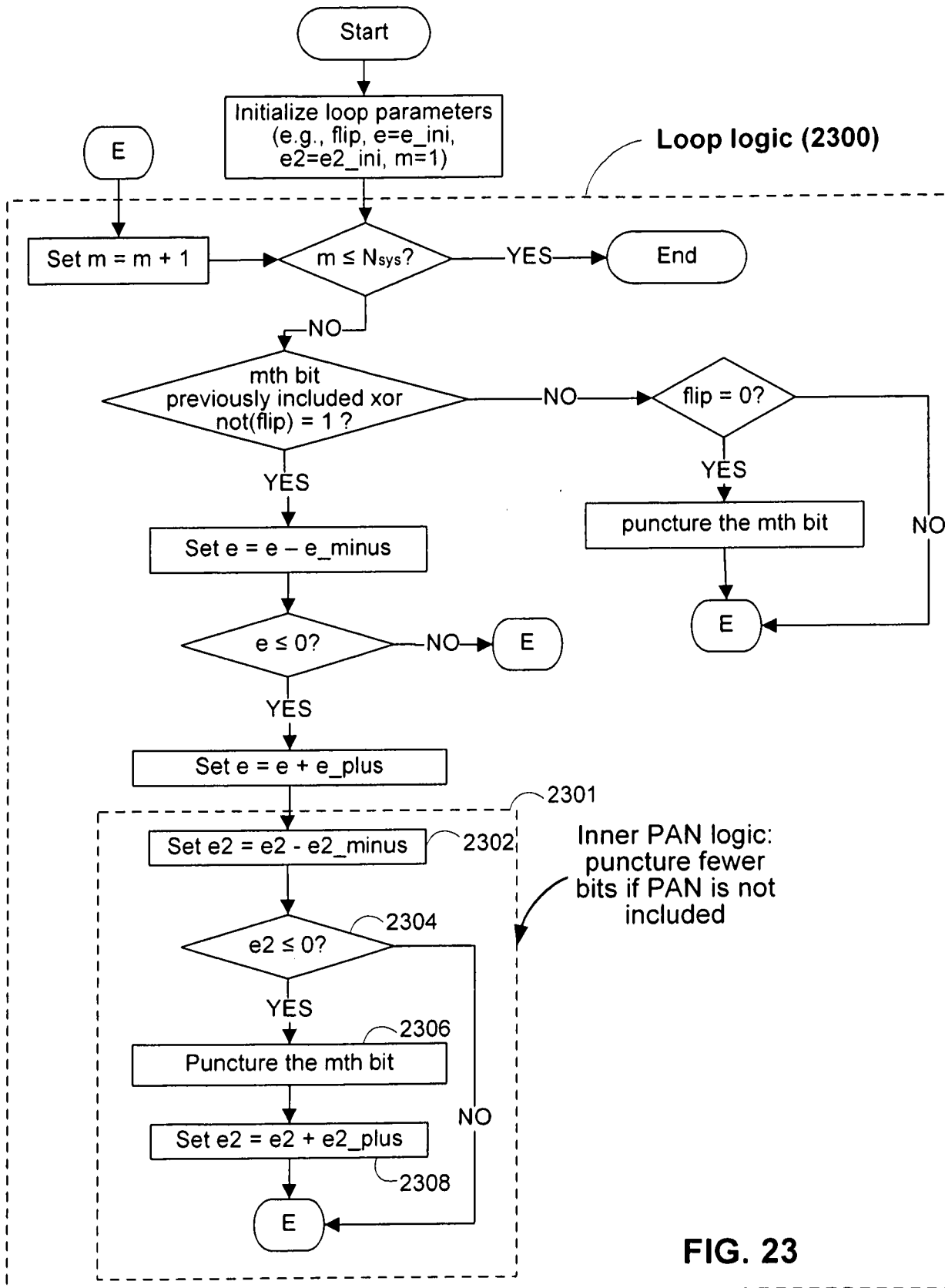
2100

FIG. 21





23/31

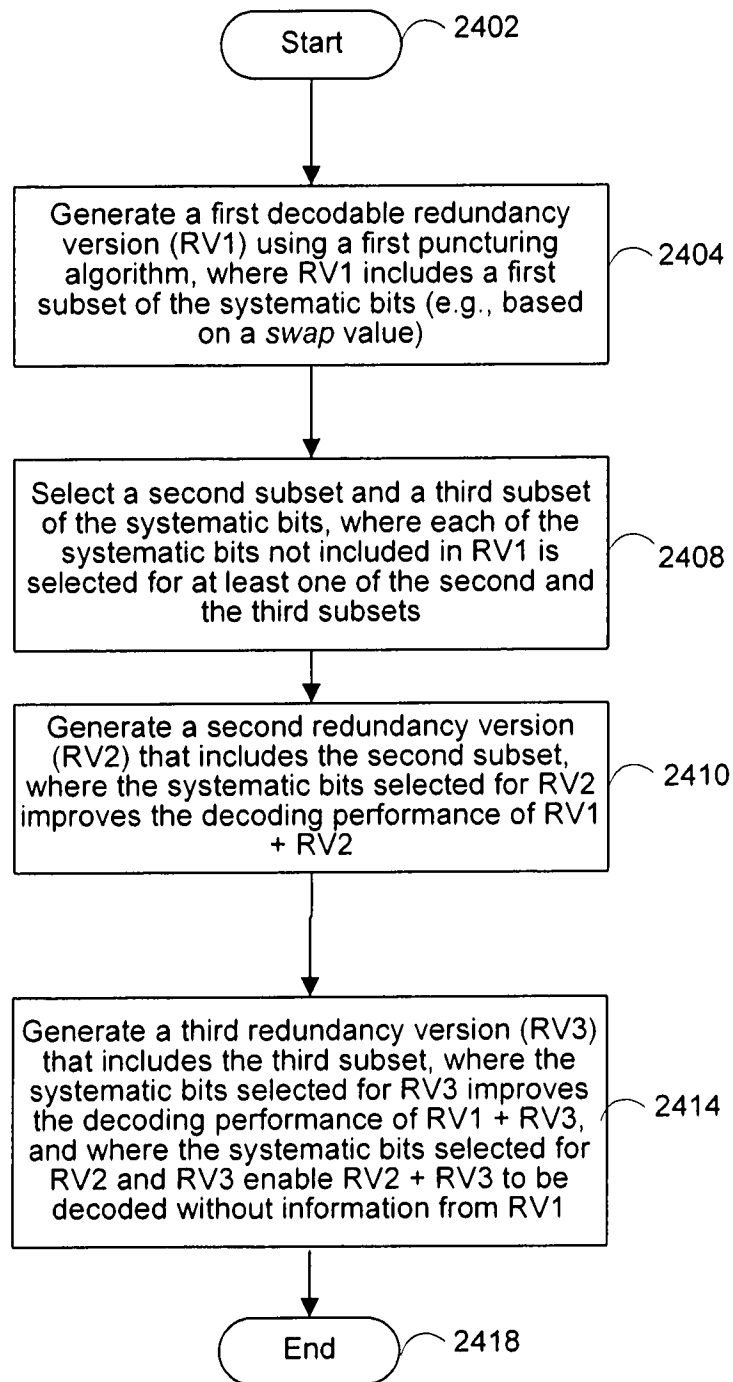
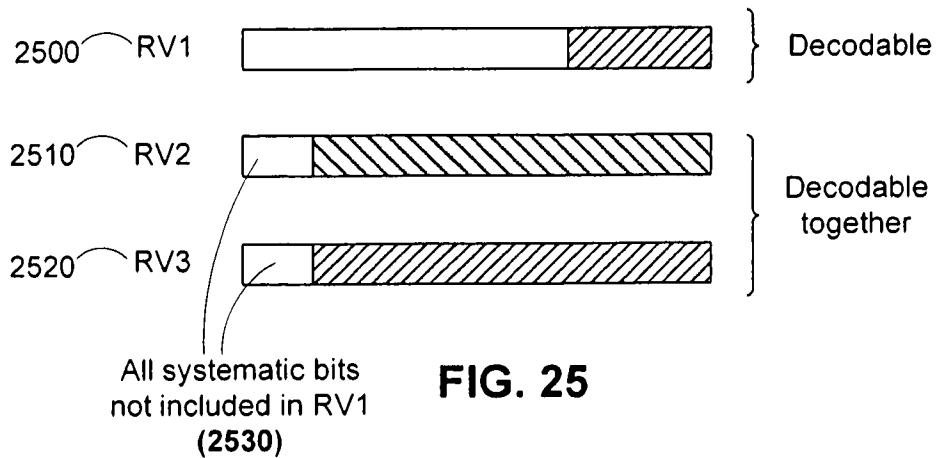
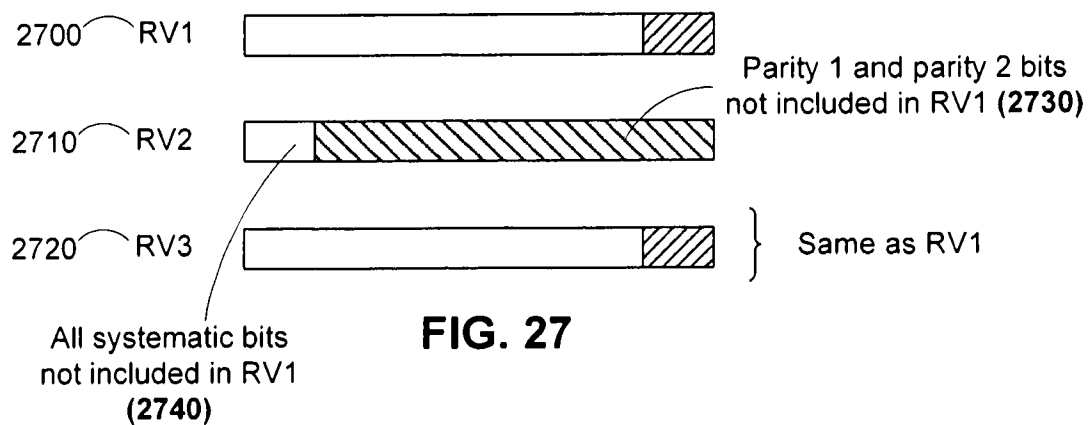
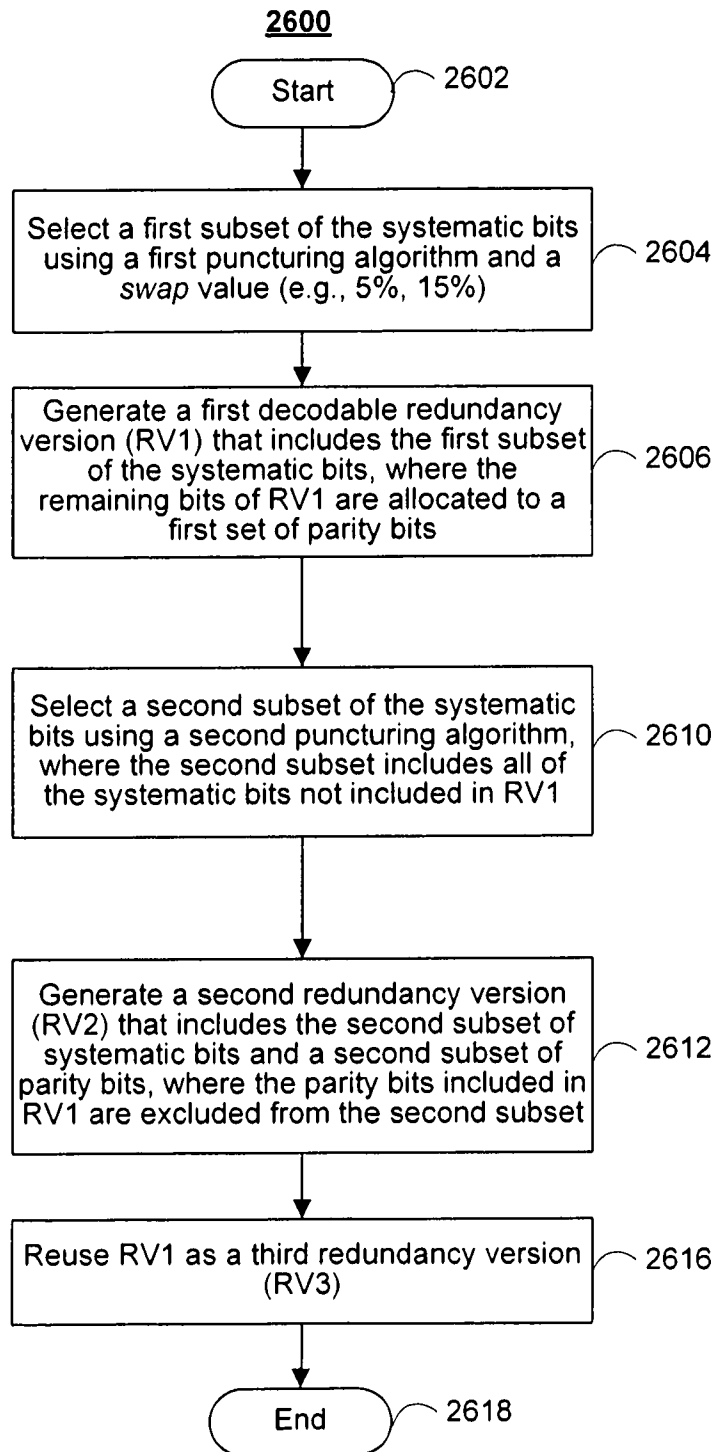
2400

FIG. 24

24/31

Example : $R = 1$ **Example : $R = 1$** 

25/31

**FIG. 26**

26/31

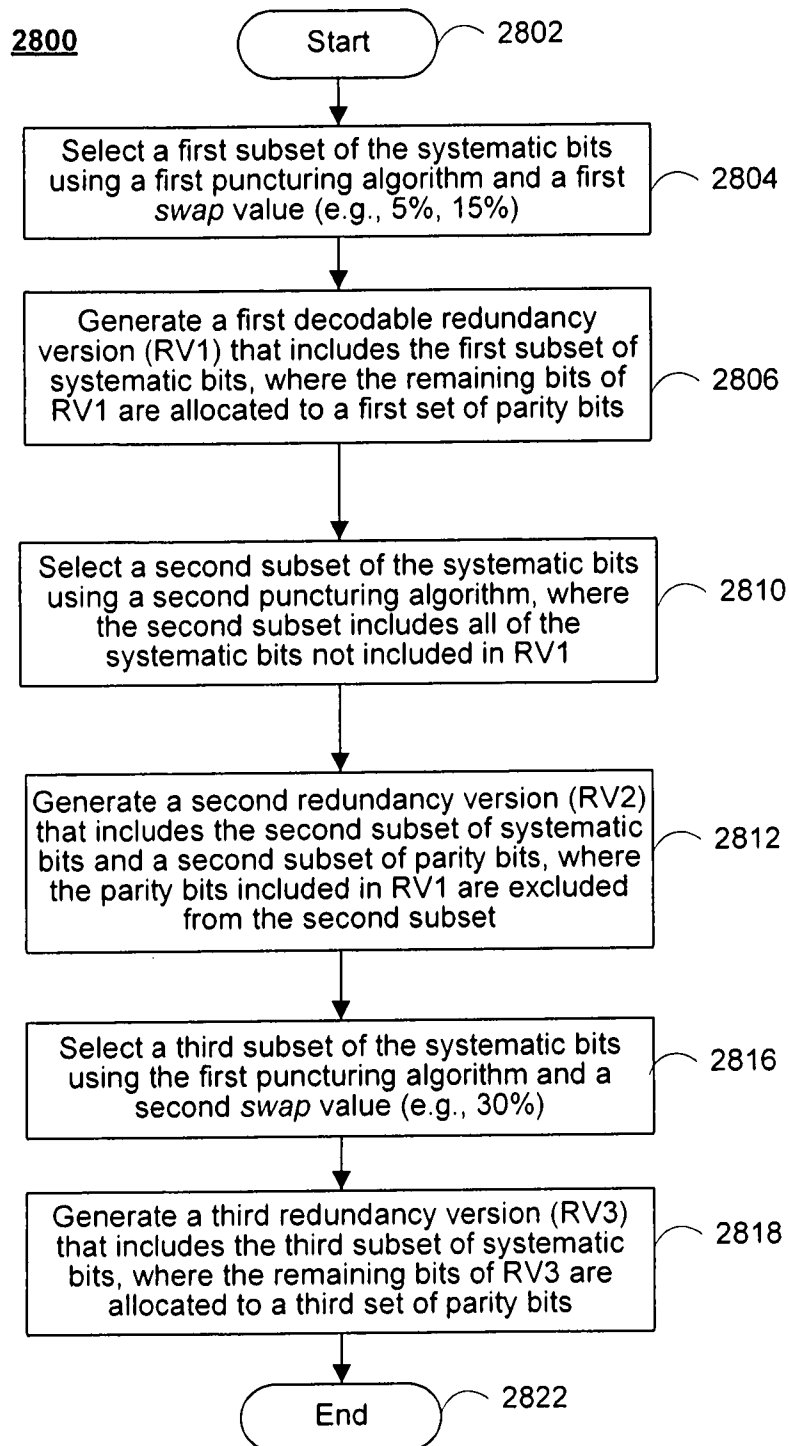
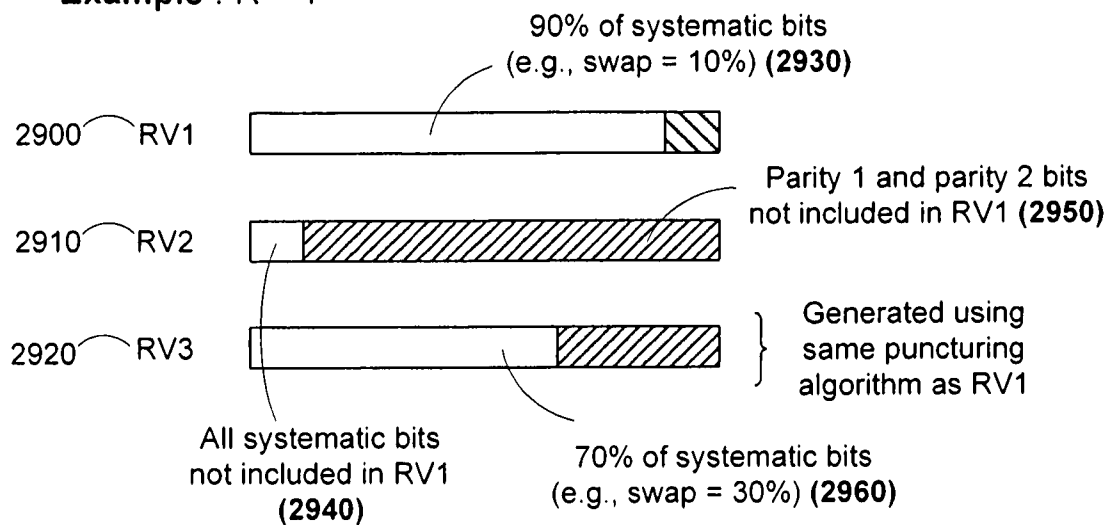


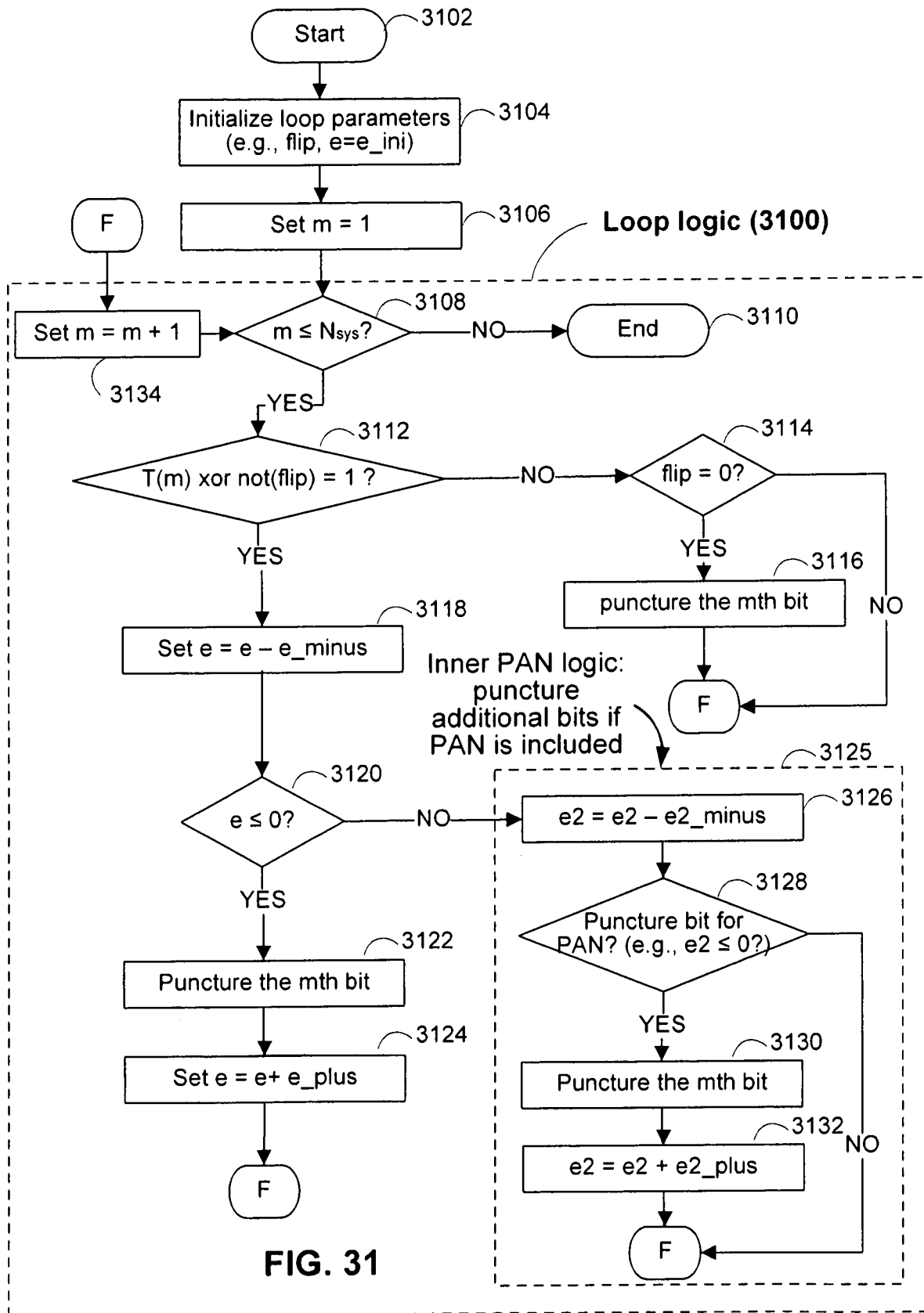
FIG. 28

27/31

Example : $R = 1$ **FIG. 29****Table 4: *Flip* values**

RV1	RV2 - I	RV2 - II	RV3
0	1	0	1

FIG. 30



29/31

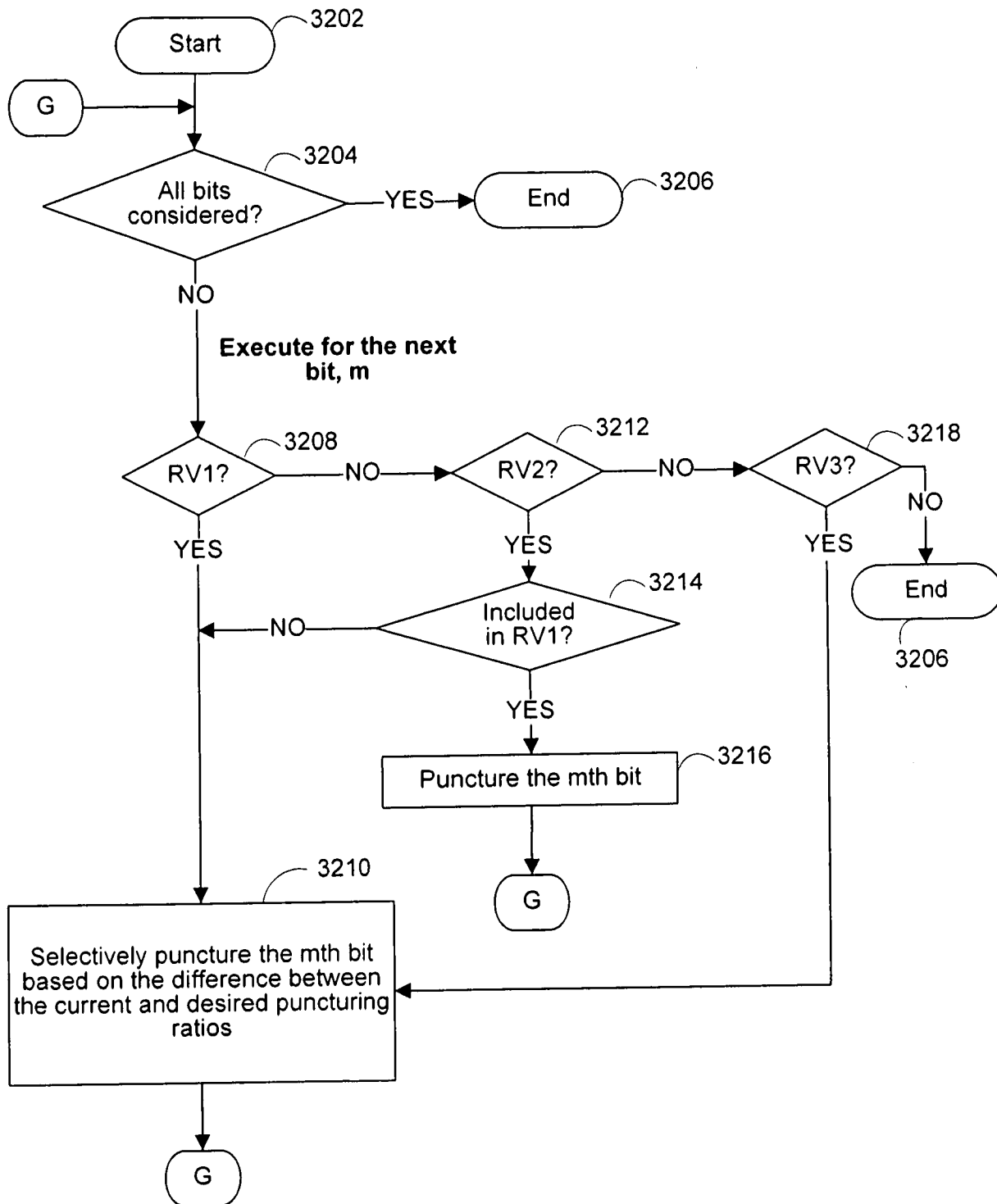
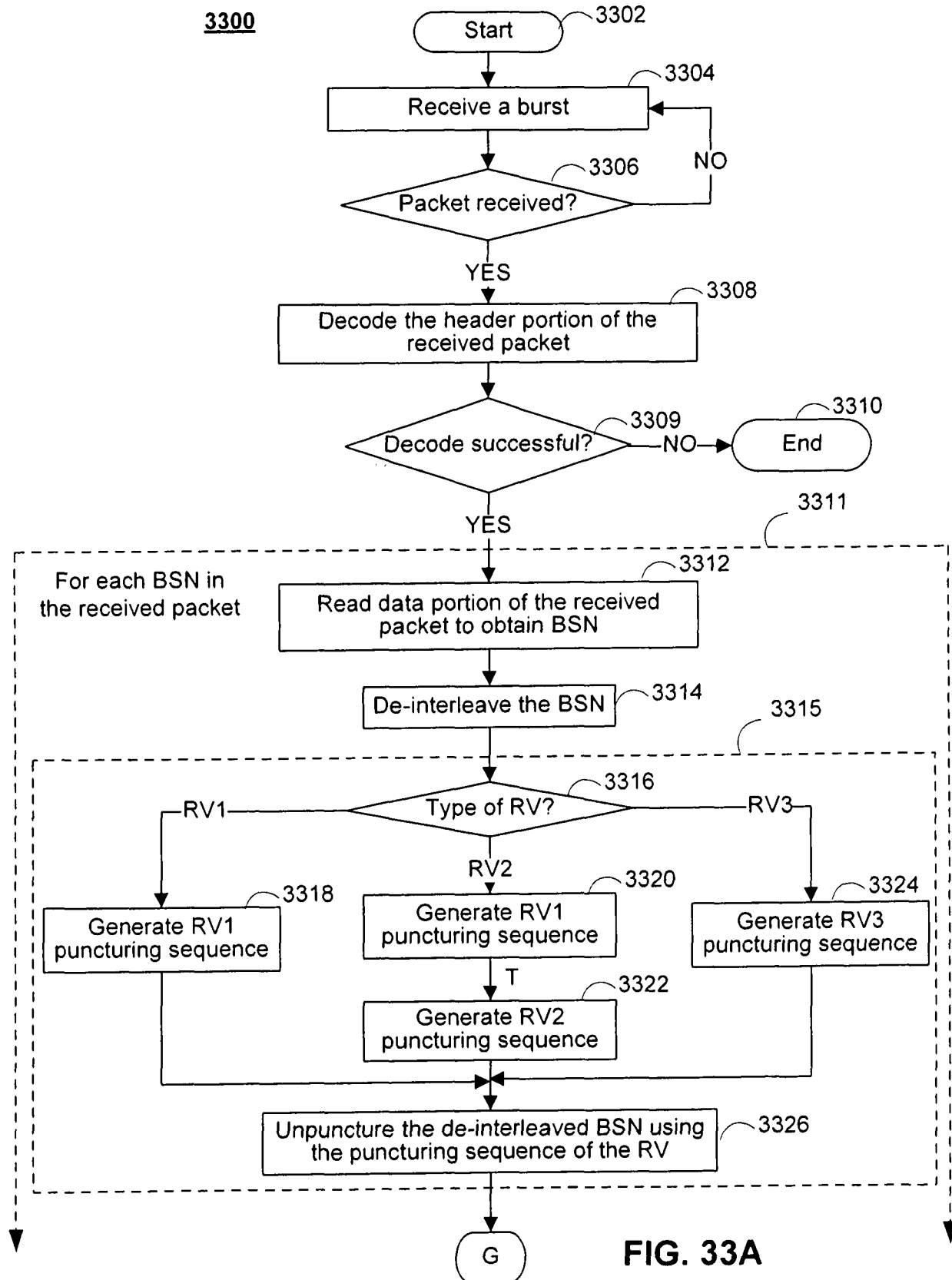
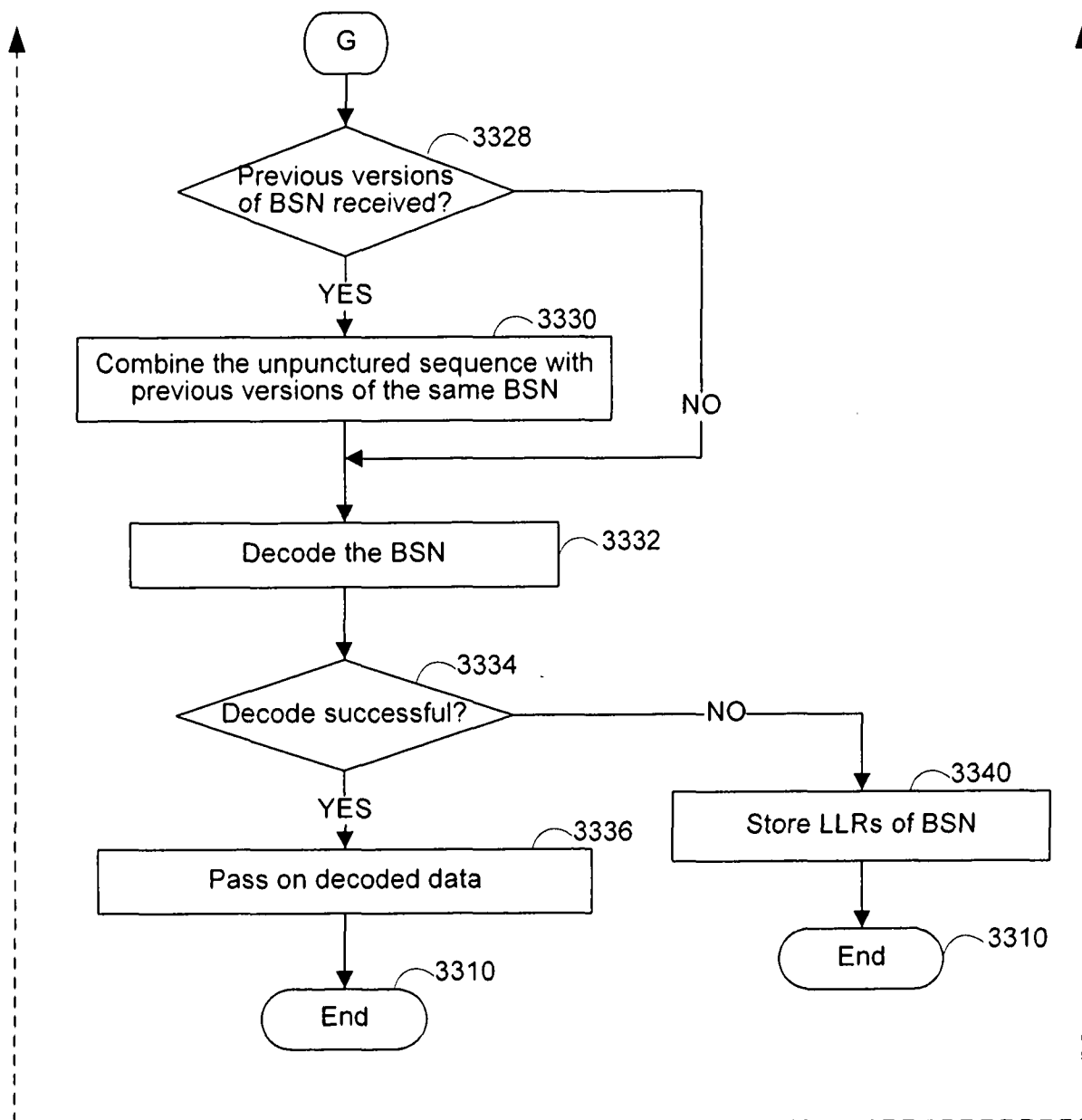
3200

FIG. 32

30/31



3300**FIG. 33B**