

(51) International Patent Classification:  
*H04L 1/00* (2006.01)(21) International Application Number:  
PCT/US2013/045748(22) International Filing Date:  
13 June 2013 (13.06.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
1210779.3 18 June 2012 (18.06.2012) GB  
13/666,774 1 November 2012 (01.11.2012) US(71) Applicant: MICROSOFT CORPORATION [US/US];  
One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: VAFIN, Renat; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). ANDERSEN, Soren Vang; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). NILSSON, Mattias; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(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, BN, BR, BW, BY, BZ, CA, CH, CL, 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, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

## Declarations under Rule 4.17:

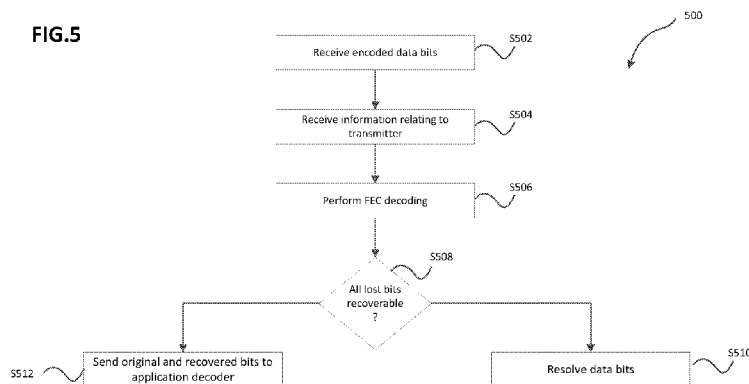
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

## Published:

- with international search report (Art. 21(3))

(54) Title: METHOD AND APPARATUS FOR ERROR RECOVERY USING INFORMATION RELATED TO THE TRANSMITTER

FIG.5



(57) Abstract: A method for processing encoded data bits transmitted from a transmitter to a receiver over a lossy communication channel, the method comprising: receiving the encoded data bits over the communication channel, the encoded data bits including redundant data units; decoding the encoded data bits at an error correction decoder, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units; determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit; receiving information relating to the transmitter; analyzing the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using information relating to the transmitter; and resolving the at least one data bit based on the analysis.

## METHOD AND APPARATUS FOR ERROR RECOVERY USING INFORMATION RELATED TO THE TRANSMITTER

### Background

[001] There are many situations in which a data stream is transmitted from a transmitter to a receiver over a communication channel. A communication  
5 channel may exist in one (or more) of many different possible networks. For example, a communication channel may exist in the internet or in a wireless network such as a mobile telecommunications network. The data in a data stream may be divided into data packets for transmission over a communication channel. The protocol for dividing the data into data packets (e.g. the form of the headers  
10 of the data packets, and other such implementation details) may depend upon the nature of the communication channel on which the data is to be transmitted, e.g. the type of network through which the data packet is to be transmitted. The data may be encoded by an encoder prior to transmission from the transmitter and decoded by a decoder after reception at the receiver.

15 [002] In an ideal system, the communication channel is lossless such that every data packet transmitted from the transmitter over the communication channel is received at the receiver. However, in real physical systems, the communication channel may be lossy, meaning that some of the transmitted data packets will be lost on the communication channel and as such will not be received at the  
20 receiver. This may be detrimental. Correction schemes can be used at the receiver to help recover at least some of the lost data packets.

[003] As an example, forward error correction (FEC) is one correction scheme which can be used to combat packet loss in a communication channel.

### Summary

25 [004] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

30 [005] Embodiments of the invention provide a method for processing encoded data bits at a receiver, wherein the encoded data bits transmitted from a transmitter to the receiver over a lossy communication channel established over a network. The encoded data bits which are received over the communication

channel include redundant data units. At an error correction decoder, the encoded data bits are decoded, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units. The method comprises a step of determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit and receiving information relating to said transmitter. The plurality of candidate bit values are analyzed to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter and the at least one data bit is resolved based on the analysis.

#### 10 Brief Description of the Drawings

[006] For a better understanding of the present invention and to show how the same may be put into effect, reference will now be made, by way of example, to the following drawings in which:

15 [007] Figure 1a illustrates a first example of how a FEC data unit may be generated;

[008] Figure 1b illustrates a second example of how a FEC data unit may be generated;

[009] Figure 1c illustrates a third example of how a FEC data unit may be generated;

20 [010] Figure 2 is a timing diagram showing the transmission of data packets in a communication system;

[011] Figure 3 shows a communication system;

[012] Figure 4 is a representation of a data stream being transmitted in a communication system;

25 [013] Figure 5 is a flow chart for a process of resolving data bits;

[014] Figure 6 is a schematic of a processor at a receiver;

[015] Figure 7 is a schematic of a processor at a transmitter.

#### Detailed Description of Preferred Embodiments

30 [016] Preferred embodiments of the invention will now be described by way of example only.

[017] While FEC reduces loss as seen by an application (e.g. audio or video) decoder compared to the loss in the communication channel, it does not guarantee recovery of all lost original packets.

[018] The extent of data transfer over public networks such as the internet is rapidly increasing. Error correction mechanisms are therefore becoming more important to ensure reliable data transmission.

5 [019] During FEC decoding uncertainties may arise as to the correct bit value or bit sequence value of a lost packet due to the FEC decoder determining that a plurality of candidate options exist for a particular bit value or bit sequence value.

[020] The inventors have realised that by evaluating the content of candidate packets some uncertainties of the recovery process (i.e. where a bit or sequence of bits could take more than one value) can be resolved to further reduce the packet loss seen by an application decoder, compared to the packet loss in the communication channel.

[021] FEC produces redundant correction data (as FEC data units) in addition to the original data units and includes the FEC data units in the data stream which is transmitted over the communication channel. A FEC scheme that transmits original and redundant FEC data is called a systematic FEC scheme. A FEC scheme that does not transmit original data is called a non-systematic scheme. In the latter case, the total amount of data transmitted is still higher than the amount of original data, to provide redundancy. For simplicity, but without loss of generality, we talk about systematic FEC schemes herein. The redundant FEC data units may be placed in their own packets, separate to the data packets in the data stream. Alternatively, or additionally, the FEC data units may be appended (or "piggybacked") to the original data packets in the data stream. For simplicity, but without loss of generality, we talk about separate original and FEC packets herein. When some of the original data packets are lost in the communication channel, the successfully arrived FEC data units and the successfully arrived data packets can be used to recover (at least some of) the lost data packets. That is, FEC reduces the packet loss seen by a decoder of the receiver, compared to the actual packet loss in the communication channel.

25 [022] Figures 1a, 1b and 1c illustrate three examples of how a FEC data unit may be generated.

[023] As illustrated in Figure 1a, a FEC data unit 104 may be generated as an exact copy of an original data packet 102 of the data stream. By including both the data packet 102 and the FEC data unit 104 in the data stream, the data in the data packet 102 is transmitted twice. Therefore if the data packet 102 is lost

during transmission but the FEC data unit 104 is successfully received then the data in the data packet 102 can be successfully recovered at the receiver (using the FEC data unit 104).

5 [024] As illustrated in Figure 1b a module 108 may be used to generate a FEC data unit 110 which is a copy of an original data packet 106 encoded at a lower bit rate. If the data packet 106 is lost during transmission but the FEC data unit 110 is successfully received then the data in the data packet 106 can be at least partially recovered at the receiver based on the FEC data unit 110. It should be noted that if the data packets 102 and 106 have the same size (e.g. the same  
10 number of bits) then the FEC data unit 110 will have a smaller size (e.g. fewer bits) than the FEC data unit 104. Therefore, although the FEC data unit 104 may be more useful in recovering the data packet 102 than the FEC data unit 110 is in recovering the data packet 106, it may be beneficial to generate FEC data units as shown in Figure 1b since the FEC data unit 110 uses less of the available bit  
15 rate on the communication channel for the data stream than the FEC data unit 104 uses.

[025] As illustrated in Figure 1c a mixing module 118 may be used to generate a FEC data unit 120 from a plurality of the original data packets, e.g. data packets 112, 114 and 116. For example the mixing module 118 may determine the bit-by-  
20 bit result of an XOR function applied to the three data packets 112, 114 and 116 to generate the FEC data unit 120. In another approach, data packets 112, 114 and 116 are combined byte-by-byte using Galois field  $GF(2^8)$  arithmetic to generate the FEC data unit 120. In this sense the FEC data unit 120 is the result of combining or mixing the original data packets 112, 114 and 116. If one of the  
25 data packets 112, 114 and 116 is lost during transmission but the other two data packets and the FEC data unit 120 are successfully received then the data in the lost data packet can be successfully recovered at the receiver (using the other two data packets and the FEC data unit 120).

[026] It can therefore be seen that FEC data units may be generated in a number  
30 of different ways. Different FEC schemes describe different ways in which the FEC data units are to be generated. A FEC scheme may describe factors relating to the generation of the FEC data units, such as: the number of FEC data units that are generated; which data packets are used to generate the FEC data units; how the FEC data units are transmitted (e.g. as separate packets or by appending

the FEC data units to the data packets); and where in the data stream the FEC data units are placed. The loss-recovery performance of a FEC scheme describes the ability of the FEC scheme to recover lost data packets at the receiver using the FEC data units.

5 [027] Generally, increasing the number of FEC data units in the data stream improves the loss-recovery performance of a FEC scheme. However, increasing the number of FEC data units in the data stream comes at a cost of bit rate efficiency.

10 [028] Loss-recovery performance of a FEC scheme largely depends on FEC overhead and FEC depth.

[029] FEC overhead describes the amount of redundant data relative to the amount of original data (e.g., the number of FEC packets relative to the number of original packets). Higher overhead generally improves the loss-recovery performance of a FEC scheme. On the other hand, higher overhead comes at a cost of bit rate efficiency. If the coding bit rate of original data is kept unchanged, then increased redundancy leads to increased total bit rate. Alternatively, if the total bit rate is constrained, then increased redundancy comes at a price of reduced coding bit rate of original data.

20 [030] FEC depth describes the displacement between the oldest and the newest of the data packets (including both of those data packets) that can be protected by (i.e. combined into) a FEC data unit. In other words, it describes the maximum number of original data packets that can be combined into a FEC data unit. Higher depth provides more possibilities of different combinations of data packets for the FEC data units, thus achieving higher flexibility of a FEC scheme, for example, to varying channel conditions. In one extreme example, where a FEC data unit is constrained to be a copy of one original data packet, higher depth provides a possibility for larger separation of the original data packet and its copy (the FEC data unit) in time. This is advantageous in bursty packet-loss conditions, since it minimizes the probability of both the original data packet and the copy (the FEC data unit) being lost in a burst on the communication channel.

30 [031] The FEC depth is related to delay in the communication. A FEC data unit combines previously-generated original data packets, and therefore there is no additional algorithmic delay at the transmitter caused by using FEC. However, in order to reconstruct a lost data packet at the receiver, we assume a delay at the

receiver, since the FEC data unit is generated and transmitted later than the data packets on which it is based. This can be seen in the timing diagram shown in Figure 2. Figure 2 shows a situation in which three of the data packets (202, 204 and 206) of the data stream are used to generate the FEC data unit 208. The FEC data unit cannot be generated until all three of the data packets 202, 204 and 206 have been generated. Therefore, the data packets 202 and 204 will be transmitted from the transmitter before the FEC data unit 208 is generated at the transmitter. Therefore, the earliest that the FEC data unit 208 can be transmitted is straight after the data packet 206. This scenario is shown in Figure 2. The transmission of the data packets and the FEC data unit takes some finite amount of time. Figure 2 depicts the loss (210) of the data packet 202 during the transmission over the lossy communication channel through the network. However, the data packets 204 and 206 and the FEC data unit 208 are all successfully received at the receiver. The data packets 204 and 206 and the FEC data unit 208 can be used to recover the lost data packet 202 at the receiver. However, the lost data packet 202 cannot be recovered until all three of: the data packets 204 and 206 and FEC data unit 208 have been received at the receiver. Therefore, as shown in Figure 2, there is a delay required for FEC recovery of the data packet 202 between the time when the data packet 202 should have been received at the receiver (if it had not been lost) and the time at which the data packet can be recovered using the FEC data unit 208. If the delay requirement for the data stream is not satisfied, the lost data packet 202 will be declared as not decodable at the receiver prior to the arrival of the FEC data unit 208. This will most likely cause the performance of the FEC scheme to drop significantly.

**[032]** However, this delay need not be necessarily introduced at the receiver by FEC as an additional delay. For example, the required delay may already exist at the receiver. One possible cause of delay at the receiver is the presence of a jitter buffer, as a jitter buffer introduces a delay into the data stream in order to mitigate random variations in packet arrival times (jitter).

**[033]** For given channel conditions, FEC overhead, and FEC depth, the selection of optimal combinations / mixings of original data packets to form the FEC data units can be driven by optimizing a selected loss-recovery performance measure. Often a number of FEC packets are jointly optimized within an optimization window.

[034] Examples of loss-recovery performance measures are:

- (i) the number of losses that can be recovered;
- (ii) the sum  $\sum p_i$ , where  $p_i$  is the non-recovery probability of the  $i$ th data packet. The sum includes original data packets within an optimization window;
- (iii) the sum  $\sum (p_i)^2$ . Again, the sum includes original data packets within an optimization window. Squaring the non-recovery probabilities punishes higher non-recovery probabilities (i.e. outliers) more heavily; and

(iv) the sum  $\sum w_i p_i$ , where  $w_i$  is a weight applied to the non-recovery probability of the  $i$ th data packet. The weights describe different levels of importance or sensitivity to loss of different data packets. Again, the sum includes original data packets within an optimization window. FEC schemes that use such a weighted performance measure are referred to as source-dependent FEC or uneven-error-protection FEC, because the protection provided to different data packets is different. This allows a greater level of protection to be provided for more important data packets of the data stream (e.g. data packets from a particular application or user or data packets having greater importance, e.g. different audio data packets and different video data packets may have different levels of protection provided to them by the FEC scheme). In this way the data packets can have different levels of priority in the FEC scheme.

[035] A communication system 300 is now described with reference to Figure 3. The communication system 300 comprises a transmitter 302, a network 308 and a receiver 310. The transmitter 302 comprises a processor 304 for processing data, a network interface 307 for the transmission (and receipt) of data across the network 308, and a memory 306 for storing data. The processor 304 may be implemented at the transmitter 302 in hardware or in software. Similarly, the receiver 310 comprises a processor 312 for processing data, a network interface 309 for receiving data (and transmission of data) across the network 308, and a memory 314 for storing data. The processor 312 may be implemented at the receiver 310 in hardware or in software.

[036] The data stream may be transmitted over a communication channel through the network 308 to the receiver 310. Methods for transmitting a data



stream over a communication channel through a network are known in the art, and as such are not described in detail herein. The receiver 310 may then process the received data packets (e.g. using the processor 312) to retrieve the data. This processing may involve depacketizing and decoding the data from the data stream. The data may be stored in the memory 314 at the receiver, or output from the receiver 310, e.g. to a user. The data in the data stream may relate to a real-time communication event (such as a voice call or a video call) between a user of the transmitter 302 and a user of the receiver 310. Alternatively, the data in the data stream may relate to a data transfer such as a file transfer between the transmitter 302 and the receiver 310. The data in the data stream may be any other suitable type of data.

**[037]** The network 308 may be any suitable network which has the ability to provide a communication channel between the transmitter 302 and the receiver 310. The network 308 may be a packet-based network. The network 308 could be a Wide Area Network (WAN) or a Local Area Network (LAN). As examples, the network 308 may be an intranet, the internet or a telephony network such as a mobile telephone network. The transmitter 302 and receiver 310 may be implemented in any suitable apparatus or device for transmitting and receiving a data stream over the network 308 respectively. For example, the transmitter 302 and receiver 310 may be implemented as user devices such as personal computers or telephones which have the ability to connect to the network 308, or as other devices such as server nodes in a network.

**[038]** Figure 4 shows a representation of a data stream 402 being transmitted in the communication system 300 from the transmitter 302 to the receiver 310. The data stream 402 includes eight data packets labelled 1 to 8 in Figure 4. Only eight data packets are shown in Figure 4 for clarity but the data stream may include many more than eight data packets. As shown in Figure 4, the data stream also includes FEC data units (labelled "FEC"). In Figure 4 the FEC data units are shown as separate packets to the data packets in the data stream. However, the FEC data units may be appended (or "piggybacked") to the data packets. When the FEC data units are appended to the data packets the FEC data units do not require their own packet formatting, such as packet headers, and as such the amount of data in the data stream required for the FEC data units may be slightly reduced. However, appending the FEC data units to the data

packets means that if a data packet is lost then so is the FEC data unit appended to that lost data packet. A FEC data unit appended to a data packet protects other data packets than the data packet to which it is appended. As explained above, in a non-systematic scheme FEC data only can be sent i.e. no original data packets of the data stream.

**[039]** As described above (e.g. in relation to Figures 1a to 1c), the FEC data units are generated based on a set of the data packets of the data stream. The set may include one or more data packet. A FEC data unit may, for example, be an exact copy of an original data packet (as illustrated in Figure 1a), or a copy of an original data packet encoded at a lower bit rate (as illustrated in Figure 1b), or a combination / mixing of original data packets (as illustrated in Figure 1c). In one approach, data packets are combined bit-by-bit using a bitwise XOR operation defined as  $(0 \text{ XOR } 0) = (1 \text{ XOR } 1) = 0$  and  $(0 \text{ XOR } 1) = (1 \text{ XOR } 0) = 1$ . In another approach, data packets are combined byte-by-byte using Galois field  $GF(2^8)$  arithmetic. FEC data units may be generated by combining two or more of the data packets.

**[040]** The communication channel through the network 308 is lossy, meaning that some of the data bits in the data stream will be lost during transmission from the transmitter 302 to the receiver 310. The FEC data units may be used at the receiver 310 to recover data bits which are lost during transmission. The term "lost original data" includes data bits which are corrupted beyond repair during transmission as well as data bits which are lost during transmission.

**[041]** As discussed above, during FEC decoding, uncertainties may arise as to the correct bit value or bit sequence value of a lost packet. This is explained further by considering an example where three packets are transmitted: original packets x1 and x2, and a FEC packet that is a mix of the two original packets,  $f1 = \text{mix}(x1, x2)$ , where mixing is based on a logic bitwise XOR function.

**[042]** If packet x1 is lost and packets x2 and f1 are received, then packet x1 can be recovered by mixing (bitwise XORing) packets x2 and f1. For example, if the bit at a certain position in x2 is equal to 1 and the bit at the same position in f1 is equal to 0, then the bit at the same position in x1 is recovered as  $(1 \text{ XOR } 0) = 1$ .

**[043]** On the other hand, if both original packets x1 and x2 are lost and only the FEC packet f1 is received, then packets x1 and x2 cannot be recovered by the FEC decoder. For example, if the bit at a certain position in f1 is equal to 0, then

there remains an uncertainty if the bits at the same position in  $x_1$  and  $x_2$  are both 0 or both 1. That is, if the bit 0 in  $f_1$  was obtained by  $(0 \text{ XOR } 0)$  or  $(1 \text{ XOR } 1)$ .

**[044]** A process of resolving these uncertainties will now be discussed with reference to Figure 5. The steps shown in Figure 5 are implemented at the  
5 processor 312 of the receiver 310.

**[045]** At step S502, the receiver 310 receives encoded data bits from a transmitter 302 over a lossy communication channel established over the network 308.

**[046]** At step S504, the receiver 310 receives information relating to the  
10 transmitter 302. The information relating to the transmitter may be represented by encoded parameter values inserted in the packets of the data stream during processing of the data to be transmitted by the processor 304. The encoded parameter values are thereby received at the receiver 310 and decoded by processor 312 at the receiver to extract the information relating to the transmitter  
15 302. Alternatively the information relating to the transmitter may be transmitted from the transmitter 302 to the receiver 310 over the network 308 separately to the data stream. It will be appreciated that the information relating to the transmitter 302 may be received at the receiver 310 without being transmitted over the network 308, for example the information relating to the transmitter 302  
20 may be input locally at the receiver 310 by a user.

**[047]** In one example, the original packets are generated by encoding a source (e.g., audio or video) by a source encoder (e.g., audio or video encoder) and packetizing the resulting bit streams. Therefore, the uncertainties may be potentially resolved by analyzing the candidate bit streams, and using the  
25 available knowledge about the source, and/or the encoding process, and/or the packetization process. In this example the information related to the transmitter may include source information (audio/video), source encoder name, source encoder mode, encoding bit rate, and packetization protocol.

**[048]** At step S506, the processor 312 performs FEC decoding on the received  
30 encoded data bits. As discussed above, the FEC decoding at step S506 does not guarantee recovery of all lost original bits due to uncertainties in the recovery process (i.e. where a bit or sequence of bits could take more than one value). Therefore at step S508 the processor 312 determines whether there are any remaining uncertainties after the FEC decoding at step S506. That is, at step

S508 the processor 312 determines whether all lost bits were recoverable following the FEC decoding at step S506. The processor 312 may implement the determining step by checking if each bit has an uncertainty or not.

5 [049] If the processor 312 determines at step S508 that all lost bits were recovered during the FEC decoding at step S506, the process proceeds to step S512. At step S512 the original and recovered bits are sent to an application decoder at the processor 312 to be decoded by the application decoder. The decoded data may be stored in the memory 314 at the receiver, or output from the receiver 310, e.g. to a user.

10 [050] Referring back to step S508, if the processor 312 determines at step S508 that all lost bits were not recovered during the FEC decoding at step S506 due to uncertainties as to the correct bit value, the process proceeds to step S510.

[051] At step S510 the processor 312 resolves the uncertainties as to the correct bit value or bit sequence value. Step 510 involves analyzing candidate options for  
15 one or more lost bit values. This analysis is implemented at processor 312 using the information relating to the transmitter 302 received at step S504.

[052] In some cases it will be possible to resolve uncertainties for individual bit values by looking at the individual bits. In other cases, the process will rely on joint analysis of bit sequence values.

20 [053] The analysis relies on the fact that the original packets are not totally random, but are generated according to certain rules. Given these rules, some original packet bit values (or bit sequence values) that could result in a given FEC packet bit value (or bit sequence values) can be deemed as impossible or as having low probabilities, and therefore be excluded as candidate options.

25 [054] The analysis of candidate options at step S510 may be implemented in a number of different ways.

[055] For example the analysis of candidate options at step S510 may include identifying invalid candidate bit sequence values.

30 [056] Alternatively or additionally the analysis of candidate options at step S510 may include identifying invalid or low probability candidate parameter values: this may be based for example on knowledge of marginal probability distributions of parameter values and may alternatively or additionally be based on correlations between different parameters. Both marginal probability distributions and

correlations between parameters may be pre-stored and/or updated in real time using successfully received/recovered bits.

5 [057] Alternatively or additionally the analysis of candidate options at step S510 may include minimizing quality degradations of the resulting decoded signal. This analysis comprises identifying if one or more of the candidate options will result in degradation in quality following subsequent decoding at an application decoder. This may be implemented by running a signal analyser or a quality estimator at the receiver 310.

10 [058] It will be appreciated that the analysis of candidate options at step S510 may include any combination of the above described methods.

[059] At step S510, candidate options (bit values or bit sequence values) which are not viable options for the lost original data can be discarded, and the uncertainties as to the correct bit value or bit sequence value in the recovery process of the lost original data may be resolved using the remaining candidate options (those not discarded).

15 [060] In some cases it will not be possible to resolve the entire bit streams of the original packets lost over the communication channel. However, in many applications it is useful to resolve at least some parts (e.g., beginnings of the bit streams), because this enables decoding of useful information / parameters. As mere examples these parameters may be perceptually-important parameters describing a speech/audio frame, or first macro blocks of a video frame.

20 [061] Although in some cases it may be sufficient to perform the bit stream analysis without running the application decoder, in many situations a major benefit can be obtained by running the application decoder on candidate bit sequences and evaluating candidate decoded parameter values.

[062] The bit stream analysis is now described in more detail with reference to Figure 6. Figure 6 is a schematic representation of processor 312 at the receiver 310. Each of the elements of the processor 312 may be implemented in hardware or in software.

30 [063] Upon receiving a data stream from the transmitter 302, an error correction decoder 602 (i.e. FEC decoder) at the receiver 310 receives the data stream.

[064] If all lost bits are recoverable (determined at step S508) then the data bits received at the receiver 310 and the recovered bits are supplied on line 605 to a first switching means 612. The first switching means 612 is controllable to supply

the data bits received at the receiver 310 and the recovered bits to an application decoder 607 to be decoded. Once decoded the application decoder 607 is arranged to output the decoded bits on line 618 via a second switching means 622. The decoded bits may be stored in the memory 314 at the receiver 310, or  
5 output from the receiver 310, e.g. to a user.

**[065]** If all lost bits are not recoverable (determined at step S508) then the data bits received at the receiver 310 and the unrecovered bits are supplied on line 603 to a bitstream analyser 608.

**[066]** The bit stream analyser 608 comprises a primary recovery block 604 and a  
10 secondary recovery block 614. The primary recovery block 604 is arranged to receive the data bits with one or more unrecoverable bits on line 603. The primary recovery block 604 is further arranged to receive information related to the transmitter (denoted  $\phi$ ) on line 606 which may include the available knowledge about the source, and/or the encoding process, and/or the packetization process  
15 and/or the rules on how the original packets are generated at the transmitter 302.

**[067]** The primary recovery block 604 is operable to attempt to resolve the uncertainties of the one or more unrecoverable bits output from the FEC decoder (received on line 603) using the information related to the transmitter ( $\phi$ ). That is, the primary recovery block 604 resolves uncertainties using bit stream analysis to  
20 identify invalid bit streams using the information related to the transmitter ( $\phi$ ).

**[068]** In some cases the primary recovery block 604 will be able to resolve all the uncertainties of the one or more unrecoverable bits. That is, primary recovery block 604 is able to implement step S510 (shown in Figure 5). In this scenario, the primary recovery block 604 is arranged to supply the data (including the resolved  
25 bits) to the first switching means 612 on line 615. The first switching means is controllable to supply the data (including the resolved bits) to the application decoder 607.

**[069]** In some cases the primary recovery block 604 will not be able to resolve all the uncertainties of the one or more unrecoverable bits received on line 603. That  
30 is, a plurality of candidate bit values will still remain for at least one bit in the data. In this scenario, the primary recovery block 604 is arranged to supply a plurality of candidate bit sequences on line 613 to the secondary recovery block 614.

**[070]** The secondary recovery block 614 comprises a first data store 610 which is arranged to store the plurality of candidate bit sequences. The secondary

recovery block 614 comprises a second data store 611 which is arranged to store decoded results from bits which have previously been decoded by the application decoder 607. The processor 312 is arranged to control the second switching means 622 to supply the bits which have been previously decoded by the application decoder 607 to the secondary recovery block 614 on line 616.

[071] The secondary recovery block 614 is arranged to supply each of the candidate bit sequences to the first switching means 612 on line 625. The first switching means is controllable to supply the candidate bit sequences to the application decoder 607. The application decoder is arranged to decode the candidate bit sequences to output decoded candidate results. The decoded candidate results output from the application decoder 607 are supplied to the secondary recovery block 614 on line 616 via the second switching means 622. Due to the uncertainties with regards to the decoded candidate results, the processor 312 controls the second switching means 622 such that the decoded candidate results output from the application decoder 607 are not output on line 618 such that they are not stored in the memory 314 at the receiver 310, or output from the receiver 310, e.g. to a user.

[072] The secondary recovery block 614 is operable to attempt to exclude decoded candidate results using the information related to the transmitter ( $\phi$ ) received on line 606, and by analyzing the decoded results which are stored in the second data store 611.

[073] The analysis can be facilitated by the fact that the ambiguous original packets (their bit streams and encoded parameters) are often correlated with each other. Also, the ambiguous packets are often correlated with earlier decoded original packets.

[074] In some cases, based on this analysis, the secondary recovery block 614 will be able to exclude all but one decoded candidate result. That is, the secondary recovery block 614 operates in addition to recovery block 604 to implement step S510 (shown in Figure 5). In this scenario the secondary recovery block 614 is arranged to output the decoded candidate result on line 617. The decoded bits output on line 617 may be stored in the memory 314 at the receiver 310, or output from the receiver 310, e.g. to a user.

[075] It will be apparent that in the bit stream analysis described above, the first switching means 612 has three input connections on lines 605, 615 and 625.

During the bit stream analysis the first switching means 612 is arranged to receive data on one of its three inputs and is controllable to supply the received data to the application decoder 607. That is, the first switching means 612 will receive one of: (i) data bits received at the receiver 310 and bits recovered by the error  
5 correction decoder 602 on line 605, (ii) bits received at the receiver 310 and bits resolved by the primary recovery block 604 on line 615, and (iii) candidate bit sequences on line 625. The data that the first switching means 612 receives on one of its three inputs depends on the success of the recovery of lost bits at the error correction decoder 602, and the success of the primary recovery block 604  
10 resolving uncertainties of one or more unrecoverable bits output from the FEC decoder.

**[076]** It will also be apparent that the architecture of the processor 312 will depend on the how the analysis of candidate options (at step S510 of Figure 5) is implemented. For example if the analysis of candidate options only comprises  
15 identifying invalid candidate bit sequence values (implemented by the primary recovery block 604) then the secondary recovery block 614 may not be required. Similarly, if the analysis of candidate options only comprises identifying invalid or low probability decoded candidate parameter values (implemented by the secondary recovery block 614) then the primary recovery block 604 may not be  
20 required. In this case, the secondary recovery block 614 will be arranged to receive the data bits with one or more unrecoverable bits output from the error correction decoder 602 and determine the candidate bit sequences that are supplied to the application decoder in the process of resolving the uncertainties of the one or more unrecoverable bits.

**[077]** In one embodiment of the invention, the transmitter 302 is aware of the receiver's 310 ability to resolve uncertainties after FEC decoding. An FEC encoder at the transmitter 302 can be optimized with the knowledge that the receiver 310 has the ability to resolve uncertainties after FEC decoding. This is discussed in further detail with reference to Figure 7.

**[078]** Figure 7 is a schematic representation of processor 304 at the transmitter 302. For simplicity the processor 304 is shown as having an FEC encoder 703, it will be appreciated that the processor may include further modules used in the transmission of data not shown in Figure 7. The FEC encoder 703 is arranged to receive encoded information (encoded data bits) output from a source encoder



(not shown in Figure 7) on line 701. The source encoder may for example be an audio or video encoder. The FEC encoder 703 is further arranged to receive information relating to the receiver ( $\beta$ ) on line 705 and output encoded data on line 707 for transmission over the network 308. In operation, the FEC encoder 703  
5 encodes the information received on line 701. That is, the FEC encoder 703 attempts to protect the source-encoded information by adding redundancy in accordance with a correction scheme that is optimized based on the information relating to the receiver ( $\beta$ ) received on line 705.

**[079]** In one embodiment of the invention the FEC optimization may be  
10 implemented by running offline "training". That is, it will be appreciated that there will be many different ways to generate the FEC data, for example whilst Figure 1c shows a FEC data unit 120 generated from three original data packets, any number of original data packets may be combined to generate a FEC data unit and therefore there is a large optimization space. By running offline simulations it  
15 is possible to determine an FEC scheme which is optimized to give the best performance according to a particular measure. As mere examples to illustrate the concept, this measure may be packet loss which can be calculated by analysing the simulation results i.e. the number of packets sent from the transmitter, the number of packets received at the receiver, the number of packets recovered by  
20 the FEC decoder and the number of packets further recovered by resolving uncertainties after FEC decoding (if this further processing is applied at the receiver). If video data is to be transmitted across the network, the measure may be a level of video quality. Video quality evaluation techniques are well known in the art and will not be discussed herein.

**[080]** These simulations may include determining the performance of each possible FEC scheme (i.e. of each permutation of how the FEC data units can be generated) over multiple realisations of the communication channel, for example the simulations may include determining the performance of each possible FEC scheme when simulating the channel suffering from various degrees of packet  
30 loss.

**[081]** The offline "training" may include running simulations where the receiver 312 comprises only a FEC decoder 602, and where the receiver 312 comprises a FEC decoder and is able to resolve remaining uncertainties after FEC decoding (having elements shown in Figure 6). It will be appreciated that the FEC scheme

that provides the best performance for the scenario where the receiver 312 comprises only a FEC decoder 602 may be different to the FEC scheme that provides the best performance for scenario where the receiver 312 comprises a FEC decoder and is able to resolve remaining uncertainties after FEC decoding.

5 [082] The FEC encoder 703 may determine the receiver's ability to resolve uncertainties after FEC decoding in a number of different ways. In one example the FEC encoder 703 is implemented by communication client software (provided by a software provider associated with the communication system 300) executed at the transmitter which allows the transmitter to engage in calls and other  
10 communication session over network 308. Where the FEC decoding (and resolving remaining uncertainties after FEC decoding) is implemented by executing communication client software at the receiver, provided by the same software provider which provides the communication client software which implements the FEC encoder 703, the FEC encoder 703 is able to determine the  
15 receiver's ability to resolve uncertainties after FEC decoding by reading the version number of the communication client software which is executed at the receiver. In another example, the FEC encoder 703 may be notified of the receiver's ability to resolve uncertainties after FEC decoding by data being fed back from the receiver 310 over the network to the transmitter 302. In yet another  
20 example, the FEC encoder 703 may determine the receiver's ability to resolve uncertainties after FEC decoding by the user manually supplying this information to the FEC encoder 703. The information relating to the receiver ( $\beta$ ) received on line 705 comprises information indicating the receiver's ability to perform FEC decoding and resolve uncertainties after FEC decoding.

25 [083] Once the FEC encoder 703 has determined whether the receiver has the ability to resolve uncertainties after FEC decoding the FEC encoder 703 is able to select the FEC scheme optimized for the receiver's capabilities.

[084] While this invention has been particularly shown and described with reference to preferred embodiments, it will be understood to those skilled in the art  
30 that various changes in form and detail may be made without departing from the scope of the invention as defined by the appendant claims.

[085] In the preferred embodiments described above FEC data is used. In alternative embodiments, other types of correction data may be used instead of

FEC data provided that the correction data allows the data stream units of a lost packet to be recovered using the correction data at the receiver.

[086] The above examples exemplify how the space of candidate bit values is formed in the case of bitwise XOR mixing. Similarly, candidate bit streams are formed for other mixing ways. For example, in Galois field  $GF(2^8)$  case the candidate bit streams are formed byte-by-byte.

[087] The above examples exemplify the remaining uncertainty in one FEC packet that represents a mix of two original packets. Similarly, the uncertainties can be jointly resolved for a higher number of FEC packets and for a higher number of original packets being mixed in FEC packets.

[088] According to a first embodiment of the invention there is provided a method of processing encoded data bits at a receiver, said encoded data bits transmitted from a transmitter to the receiver over a lossy communication channel established over a network, the method comprising: receiving the encoded data bits over the communication channel, the encoded data bits including redundant data units; decoding the encoded data bits at an error correction decoder, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units; determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit; receiving information relating to said transmitter over the network; analyzing the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter; and resolving the at least one data bit based on said analysis.

[089] The information relating to said transmitter may be received over the network.

[090] When the step of decoding operates on sequences of data bits and finds a plurality of candidate bit values for each data bit, the method may further comprise analyzing the plurality of candidate bit values for each bit in a sequence using the information relating to said transmitter, to determine most likely candidate bit values for the sequence.

[091] Preferably, the information relating to said transmitter includes at least one of: information on a source of the data bits prior to being encoded; information on an encoding process used to encode the encoded data bits; information on a packetization process used to packetize the encoded data bits; and information on

predetermined rules followed by the transmitter in generating the encoded data bits.

5 [092] The analyzing may further comprise identifying if one or more of the plurality of candidate bit values for the at least one data bit will result in a degradation in quality following a subsequent decoding step.

[093] Preferably, the step of analyzing comprises decoding the candidate bit values using an application decoder to generate a candidate decoded result, and using the information related to said transmitter to exclude invalid or low probability candidate bit values based on the candidate decoded result.

10 [094] The method may further comprise the application decoder decoding recovered or resolved data bits to generate decoded data bits; and storing the decoded data bits for use in excluding invalid or low probability candidate decoded results.

15 [095] The method may further comprise selectively supplying candidate bit values to the application decoder to decode the candidate bit values when candidate bit values cannot be resolved using the information relating to said transmitter.

20 [096] The candidate decoded result may represent a parameter of an original data unit, the analyzing may comprise excluding candidate decoded results representing invalid or low probability values for the parameter based on at least one of: knowledge of marginal probability distributions of parameter values; and correlations between different parameter values.

[097] The error correction decoder may implement a forward error correction scheme.

25 [098] The encoded data bits received at the receiver may further include original data units according to a systematic correction scheme.

30 [099] When the encoded data bits received at the receiver further include original data units according to a systematic correction scheme, the encoded data bits may include redundant data units which have been generated from the original data units in one of the following ways: i) as an exact copy of the original data units; or ii) as a copy of the original data units encoded at the transmitter at a lower bit rate than the original data units; or iii) by combining a plurality of original data units.

[0100] When the redundant data units have been generated from the original data units by combining a plurality of original data units, the original data units may be combined bit-by-bit using a bitwise XOR operation or byte-by-byte using Galois field GF ( $2^8$ ) arithmetic.

5 [0101] The redundant data units received at the receiver may represent original data and redundant data according to a non-systematic correction scheme.

[0102] The network may be a packet based network.

[0103] According to a second embodiment of the invention there is provided a receiver for processing encoded data bits, said encoded data bits transmitted from  
10 a transmitter to the receiver over a lossy communication channel established over a network, the receiver comprising: receiving means for receiving the encoded data bits over the communication channel, the encoded data bits including redundant data units; an error correction decoder for decoding the encoded data bits, wherein the recovery of lost data is implemented at the error correction  
15 decoder using at least one of the redundant data units; determining means for determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit; receiving means for receiving information relating to said transmitter; analyzing means for analyzing the plurality of candidate bit values to exclude at least one of the  
20 candidate bit values for the at least one data bit using the information relating to said transmitter; and resolving means for resolving the at least one data bit based on said analysis.

[0104] The receiver may further comprise an application decoder configured to decode recovered or resolved data bits to generate decoded results and  
25 selectively operable to decode candidate bit values to generate candidate decoded results for use by the analyzing means and the resolving means.

[0105] The receiver may further comprise switching means for supplying candidate bit values to the application decoder.

[0106] The application decoder may be a voice decoder or a video decoder.

30 [0107] According to a third embodiment of the invention there is provided a method of processing encoded data bits at a receiver, said encoded data bits transmitted from a transmitter to the receiver over a lossy communication channel established over a network, the method comprising: receiving the encoded data bits over the communication channel, the encoded data bits contained in packets

and redundant data units; decoding the encoded data bits at an error correction decoder, wherein the recovery of lost packets is implemented at the error correction decoder using at least one of the redundant data units and at least one other packet; determining if at least one lost packet is unable be recovered due to the decoder finding a plurality of candidate bit values for at least one data bit in the lost packet; receiving information relating to said transmitter over the communication channel; analyzing the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter; and resolving one or more data bits of the at least one lost packet based on said analysis.

**[0108]** According to a fourth embodiment of the invention there is provided a method of processing data bits at a transmitter for transmission from the transmitter to a receiver over a lossy communication channel established over a network, the method comprising: receiving, at an error correction encoder, the data bits from a source encoder; encoding the data bits at the error correction encoder to output encoded data bits, the encoded data bits contained in packets and redundant data units in accordance with a correction scheme; and receiving at the error correction encoder information relating to the receiver, said information comprising an indication that the receiver comprises an error correction decoder for decoding the encoded data bits, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units, and that the receiver is arranged to: determine if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit; receive information relating to said transmitter; analyze the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter; and resolve the at least one data bit based on said analysis; wherein the correction scheme used by the error correction encoder is optimized based on the information relating to the receiver.

**[0109]** According to a fifth embodiment of the invention there is provided a transmitter for processing data bits for transmission from the transmitter to a receiver over a lossy communication channel established over a network, the transmitter comprising: an error correction encoder arranged to receive the data bits from a source encoder and output encoded data bits, the encoded data bits

- contained in packets and redundant data units in accordance with a correction scheme, the error correction encoder comprising: receiving means for receiving information relating to the receiver, said information comprising an indication that the receiver comprises an error correction decoder for decoding the encoded data
- 5 bits, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units, and that the receiver is arranged to: determine if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit; receive information relating to said transmitter; analyze the plurality of candidate
- 10 bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter; and resolve the at least one data bit based on said analysis; wherein the correction scheme used by the error correction encoder is optimized based on the information relating to the receiver.
- 15 **[0110]** According to a sixth embodiment of the invention there is provided a communication system comprising a receiver and a transmitter described hereinabove.

Claims

1. A method of processing encoded data bits at a receiver, said encoded data bits transmitted from a transmitter to the receiver over a lossy communication channel established over a network, the method comprising:
  - 5 receiving the encoded data bits over the communication channel, the encoded data bits including redundant data units;  
decoding the encoded data bits at an error correction decoder, wherein the recovery of lost data is implemented at the error correction decoder using at least one of the redundant data units;
  - 10 determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit;  
receiving information relating to said transmitter;  
analyzing the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to  
15 said transmitter; and  
resolving the at least one data bit based on said analysis.
2. A method according to claim 1, wherein the information relating to said transmitter is received over the network.
3. A method according to claim 1 or 2, wherein when the step of decoding  
20 operates on sequences of data bits and finds a plurality of candidate bit values for each data bit, the method further comprising analyzing the plurality of candidate bit values for each bit in a sequence using the information relating to said transmitter, to determine most likely candidate bit values for the sequence.
4. A method according to any preceding claim, wherein the information  
25 relating to said transmitter includes at least one of:  
information on a source of the data bits prior to being encoded;  
information on an encoding process used to encode the encoded data bits;  
information on a packetization process used to packetize the encoded data bits; and  
30 information on predetermined rules followed by the transmitter in generating the encoded data bits.
5. A method according to any preceding claim, wherein said analyzing comprises identifying if one or more of the plurality of candidate bit values for the



at least one data bit will result in a degradation in quality following a subsequent decoding step.

6. A method according to any preceding claim, wherein the step of analyzing comprises decoding the candidate bit values using an application decoder to  
5 generate a candidate decoded result, and using the information related to said transmitter to exclude invalid or low probability candidate bit values based on the candidate decoded result.

7. A method according to claim 6, the method further comprising:  
the application decoder decoding recovered or resolved data bits to  
10 generate decoded data bits; and  
storing the decoded data bits for use in excluding invalid or low probability candidate decoded results.

8. A method according to claim 6 or 7, the method further comprising:  
selectively supplying candidate bit values to the application decoder to  
15 decode the candidate bit values when candidate bit values cannot be resolved using the information relating to said transmitter.

9. A method according to any of claims 6 to 8, wherein the candidate decoded result represents a parameter of an original data unit, and said analyzing comprises excluding candidate decoded results representing invalid or low  
20 probability values for the parameter based on at least one of knowledge of: marginal probability distributions of parameter values; and correlations between different parameter values.

10. A receiver for processing encoded data bits, said encoded data bits transmitted from a transmitter to the receiver over a lossy communication channel  
25 established over a network, the receiver comprising:

receiving means for receiving the encoded data bits over the communication channel, the encoded data bits including redundant data units;

an error correction decoder for decoding the encoded data bits, wherein the recovery of lost data is implemented at the error correction decoder using at  
30 least one of the redundant data units;

determining means for determining if at least one data bit is unable to be recovered due to the decoder finding a plurality of candidate bit values for the at least one data bit;

receiving means for receiving information relating to said transmitter;

analyzing means for analyzing the plurality of candidate bit values to exclude at least one of the candidate bit values for the at least one data bit using the information relating to said transmitter; and

5 resolving means for resolving the at least one data bit based on said analysis.

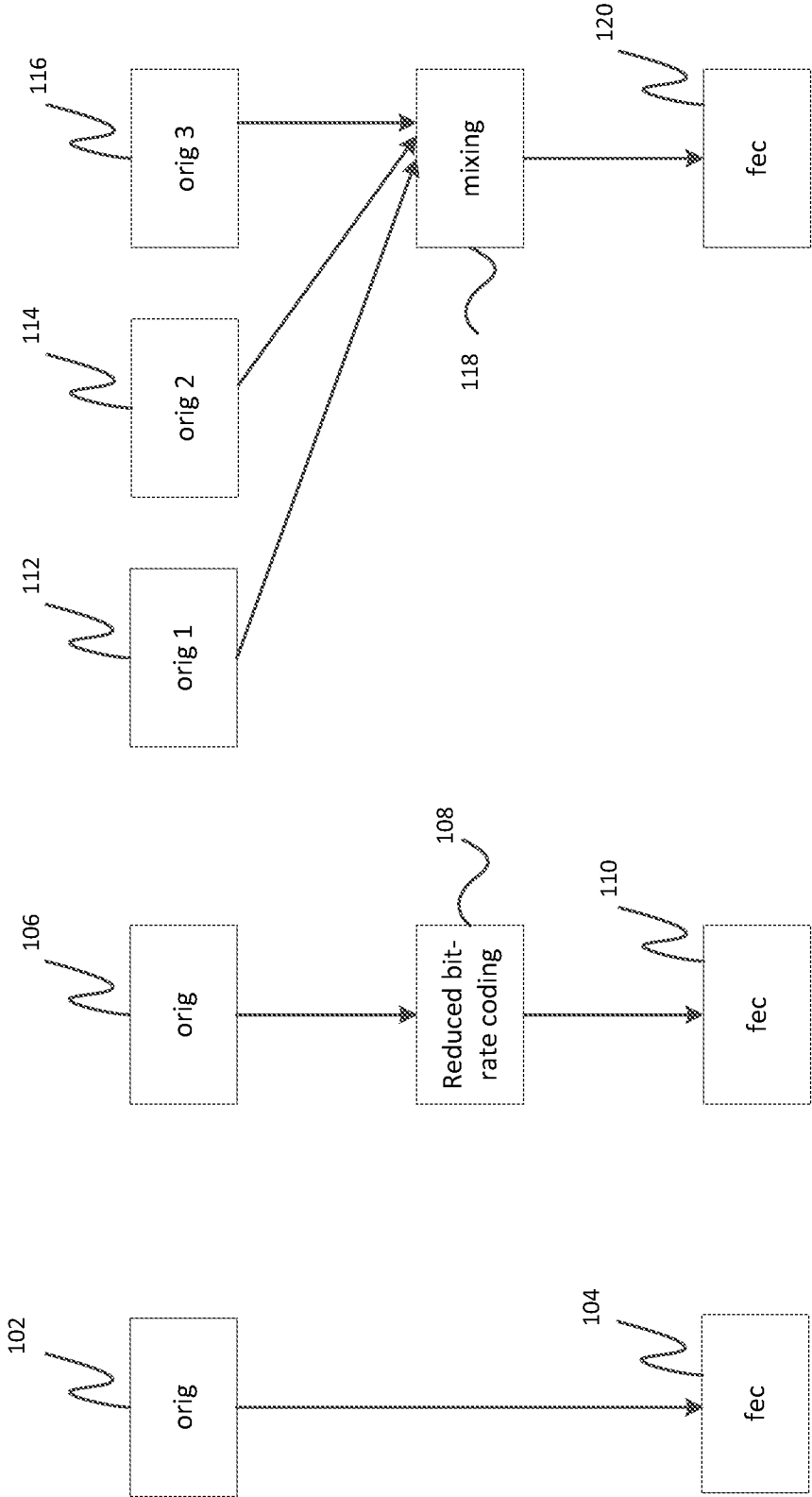


FIG.1a

FIG.1b

FIG.1c

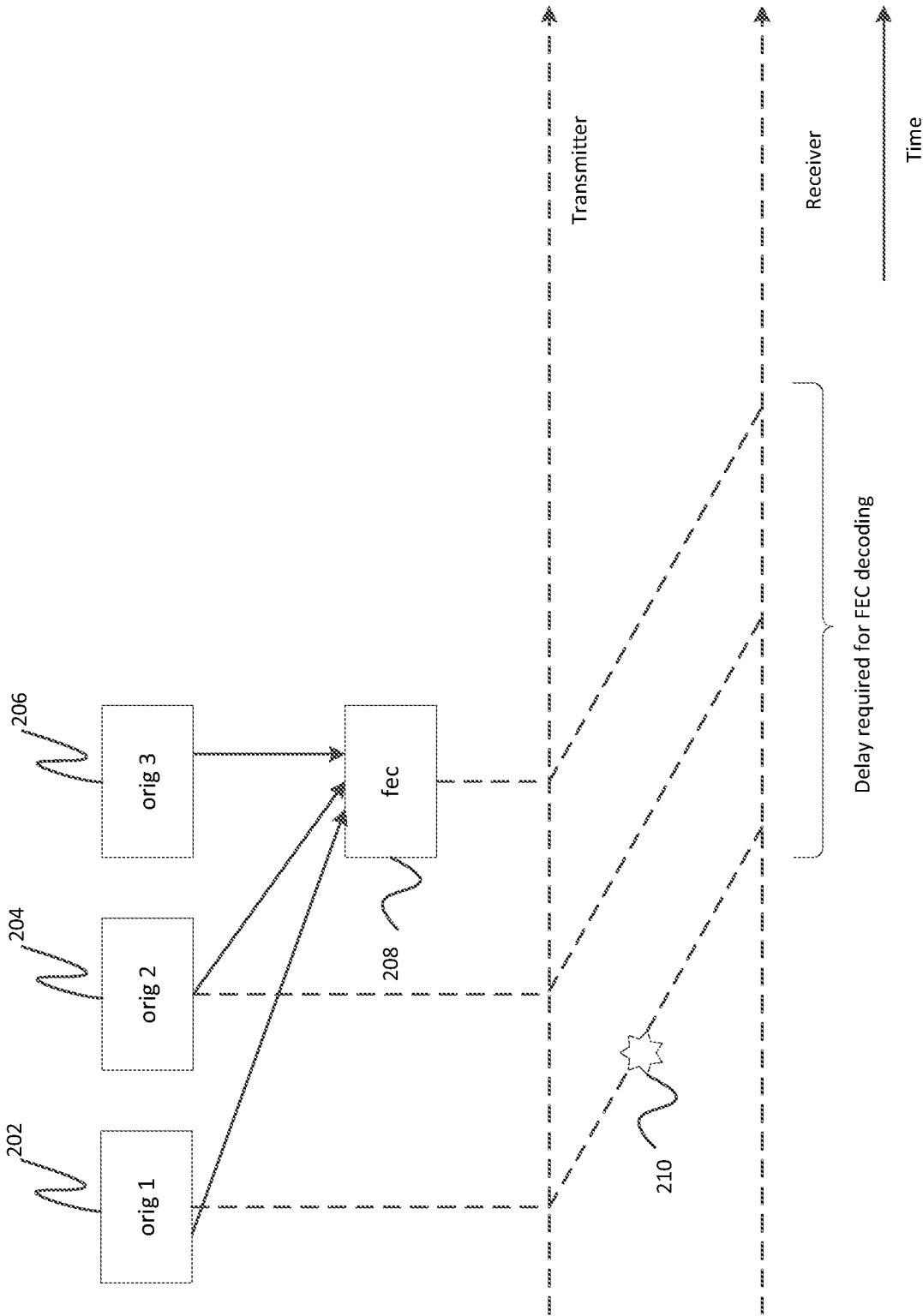


FIG.2

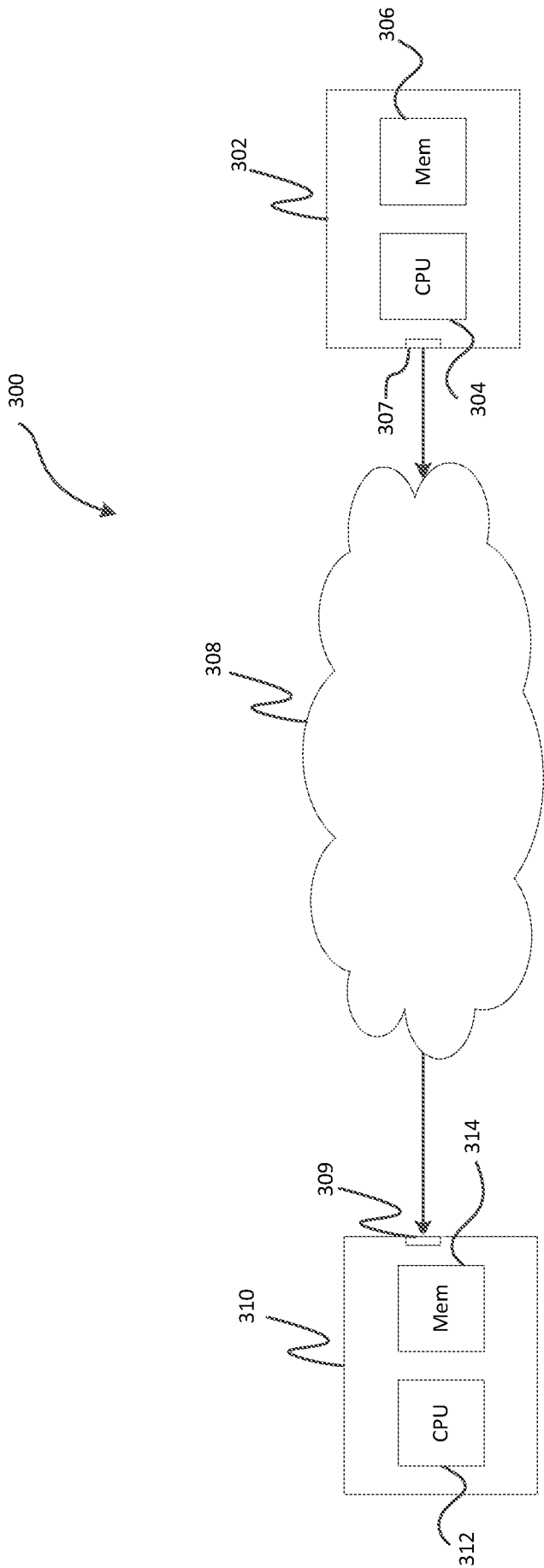


FIG.3

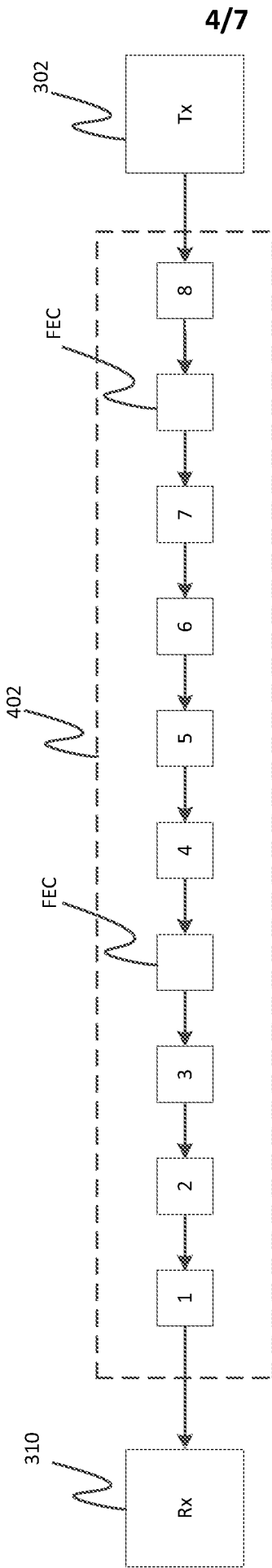


FIG.4

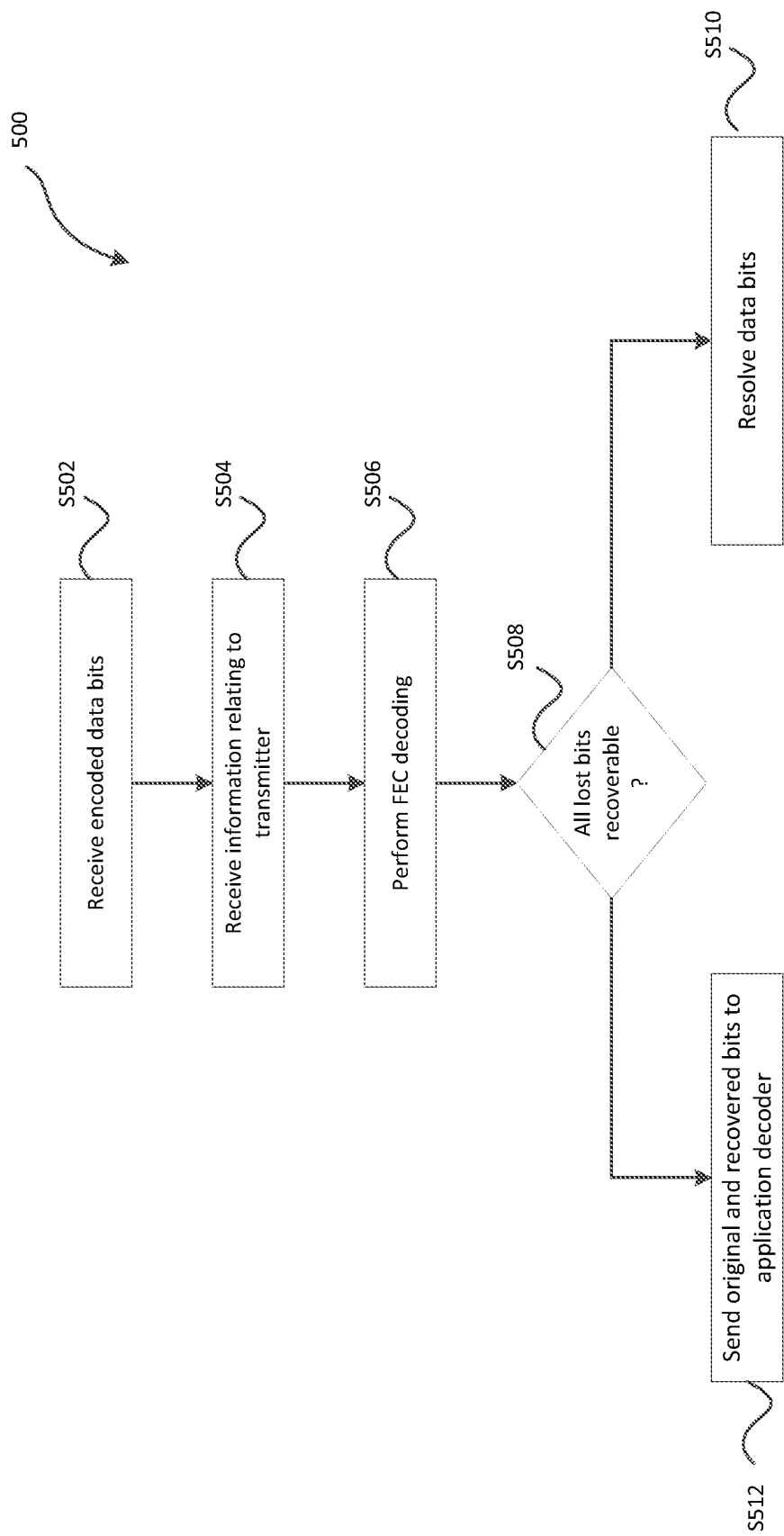


FIG.5

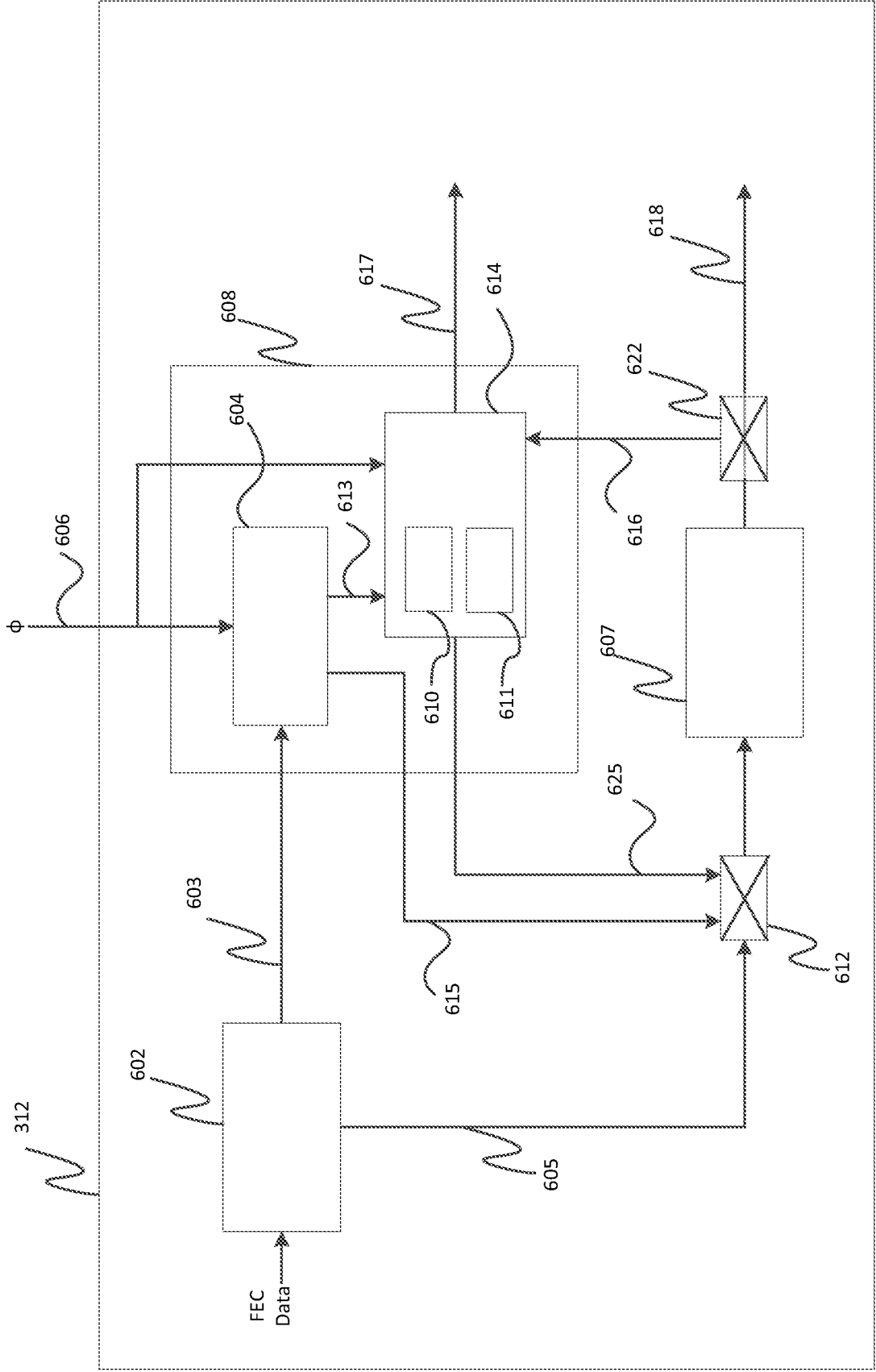


FIG. 6



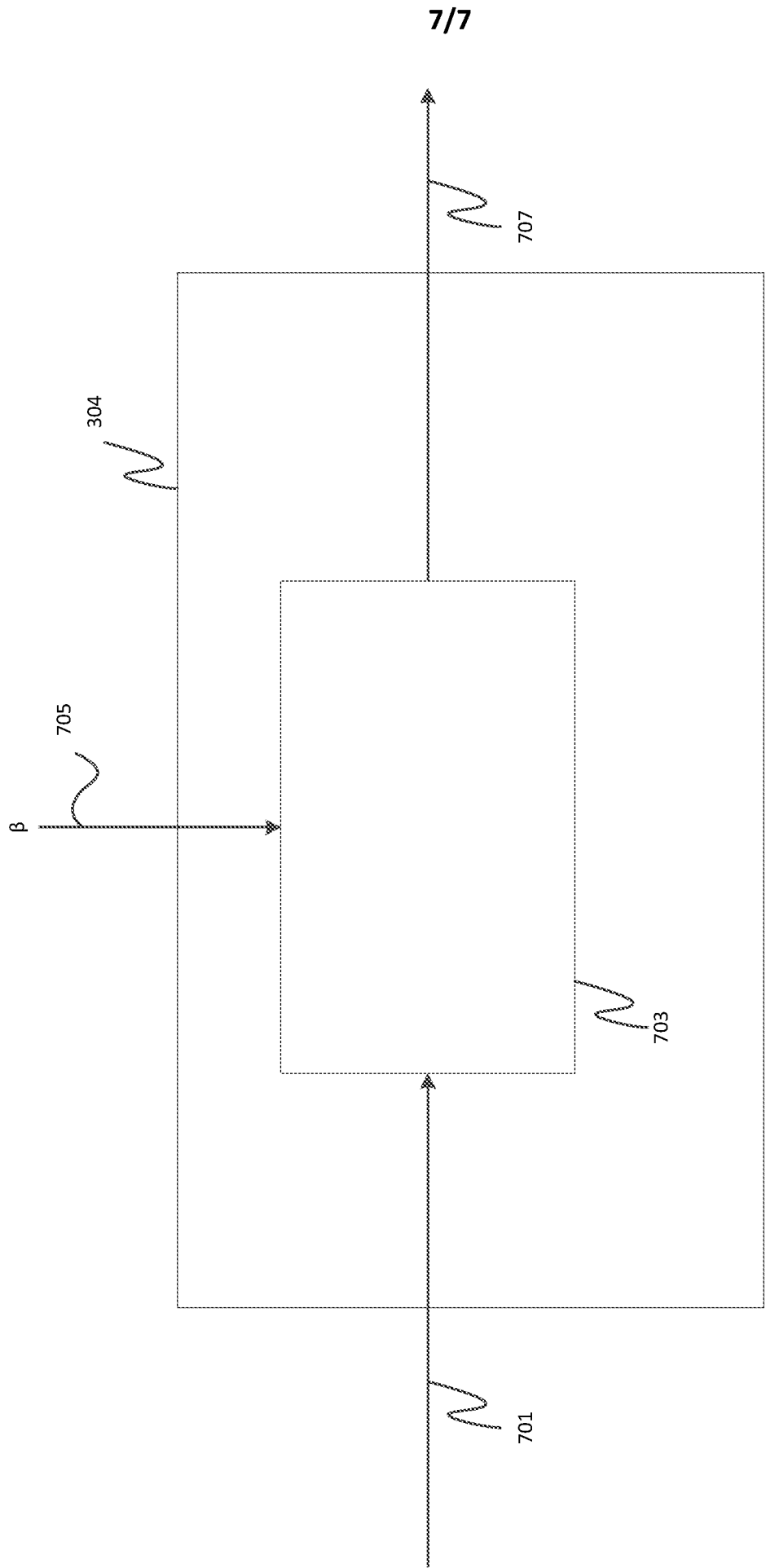


FIG.7

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2013/045748

A. CLASSIFICATION OF SUBJECT MATTER  
INV. H04L1/00  
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, COMPENDEX, INSPEC, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2 117 149 A1 (ALCATEL LUCENT [FR]; ECOLE SUPERIEURE ELECTRICITE [FR]; CENTRE NAT REC) 11 November 2009 (2009-11-11) paragraphs [0001], [0002], [0005], [0026] - [0041], [0072], [0086], [0136] - [0141]	1-10
A	----- US 2009/316823 A1 (LEE CHUN WOO [US] ET AL) 24 December 2009 (2009-12-24) paragraphs [0008] - [0010], [0028], [0056] - [0084]	1-10
A	----- US 2009/175389 A1 (HUANG QIANG [US] ET AL) 9 July 2009 (2009-07-09) paragraphs [0010], [0024] - [0026], [0054] - [0072]	1-10
	-----	

☐

Further documents are listed in the continuation of Box C.

☒

See patent family annex.

\* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

19 August 2013

Date of mailing of the international search report

26/08/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer

Miclea, Sorin

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2013/045748

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2117149	A1	11-11-2009	AT 531153 T 15-11-2011
			CN 102017497 A 13-04-2011
			EP 2117149 A1 11-11-2009
			JP 2011520372 A 14-07-2011
			KR 20110030423 A 23-03-2011
			US 2011083032 A1 07-04-2011
			WO 2009135882 A1 12-11-2009
-----			
US 2009316823	A1	24-12-2009	AT 533267 T 15-11-2011
			CA 2726123 A1 23-12-2009
			CN 102047624 A 04-05-2011
			EP 2304909 A2 06-04-2011
			JP 2011526437 A 06-10-2011
			KR 20110030623 A 23-03-2011
			RU 2011101711 A 27-07-2012
			TW 201004210 A 16-01-2010
			US 2009316823 A1 24-12-2009
			WO 2009155383 A2 23-12-2009
-----			
US 2009175389	A1	09-07-2009	TW 201006139 A 01-02-2010
			US 2009175389 A1 09-07-2009
			WO 2010009232 A2 21-01-2010
-----			