

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2011/114191 A1

(43) International Publication Date
22 September 2011 (22.09.2011)

- (51) International Patent Classification:
H04L 12/56 (2006.01)
- (21) International Application Number:
PCT/IB2010/05 1208
- (22) International Filing Date:
19 March 2010 (IS'03.2010)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).
- (71) Applicant (for LC only): **NOKIA, INC.** [US/US]; 102 Corporate Park Drive, White Plains, New York 10604 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **HEIDE, Janus** [DK/DK]; Magisterparken 191,4, DK-9000 Aalborg (DK). **PEDERSEN, Morten** [DK/DK]; Slotsgade 21,1,2, DK-9000 Aalborg (DK). **FITZEK, Frank** [DK/DK]; Jyllandsgade 21c, DK-9000 Aalborg (DK). **ZHANG, Qi** [DK/DK]; Borgmester Fischers vej 5F,2.3, DK-2000 Frederiksberg (DK).

- (74) Agents: **HARRINGTON, Mark F.** et al; Harrington & Smith, Attorneys at Law, LLC, 4 Research Drive, Shelton, Connecticut 06484-6212 (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, 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, 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, PE, PG, PH, PL, PT, RO, RS, RU, 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, 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, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(54) Title: INTRINSIC INFORMATION CONVEYANCE IN NETWORK CODING

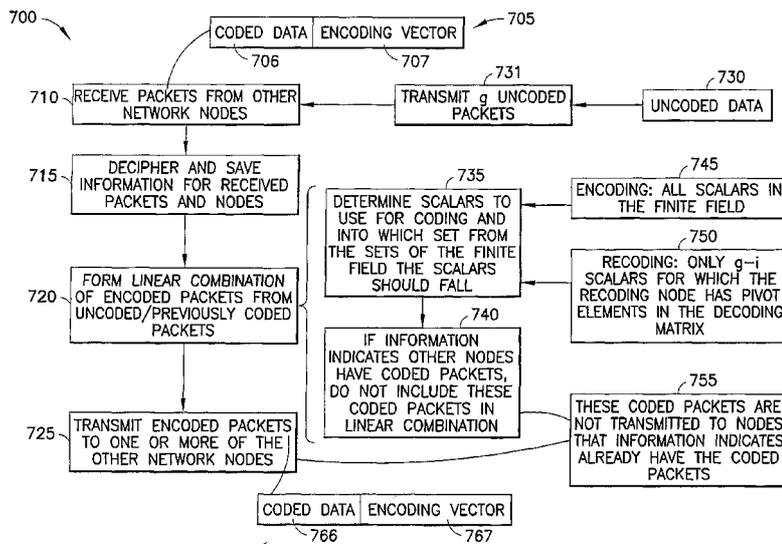


FIG. 7

(57) Abstract: A method includes receiving a first packet from at least one network node, the first packet including an encoding vector, and, based on at least the encoding vector, determining a status of the first packet at the network node. The method includes, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a number of second packets, and encoding the number of second packets to create a number of encoded second packets, wherein based on the decision the number of second packets includes or does not include the first packet. The method further includes transmitting the number of encoded second packets to the at least one network node. Apparatus and program products are also disclosed.

WO 2011/114191 A1

INTRINSIC INFORMATION CONVEYANCE IN NETWORK CODING

TECHNICAL FIELD

[0001] This invention relates generally to wired and wireless networks and, more specifically, relates to network coding in wired and wireless networks.

BACKGROUND

[0002] Network coding (NC) breaks with the current paradigm in switching computer networks often referred to as "store and forward" or "routing". In a nutshell, NC allows data flows from one or several sources to one or several sinks to be combined within a network and provides the means to subsequently separate these flows and thus reconstruct the original data at the sinks. NC is generally introduced into a network by adding a new layer of coding between the source on the application layer and channel coding on the physical layer. This new layer of coding enables a node in the network to encode, recode and decode data. Encoding and decoding are known from end-to-end codes, but recoding in the network is what separates NC from other types of coding.

[0003] While NC has certain benefits, NC can be improved.

BRIEF SUMMARY

[0004] In one exemplary embodiment, a method includes receiving a first packet from at least one network node, the first packet including an encoding vector, and, based on at least the encoding vector, determining a status of the first packet at the network node. The method includes, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a number of second packets, and encoding the number of second packets to create a number of encoded second packets, wherein based on the decision the number of second packets includes or does not include the first packet. The method further includes transmitting the number of encoded second packets to the at least one network node.

[0005] In another exemplary embodiment, a computer program is disclosed that includes code for receiving a first packet from at least one network node, the first packet comprising an encoding vector; code for, based on at least the

encoding vector, determining a status of the first packet at the network node; code for, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; code for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and code for causing transmission of the plurality of encoded second packets to the at least one network node; when the computer program is run on a processor.

[0006] In a further exemplary embodiment, an apparatus is disclosed that includes at least one processor; and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following: receiving a first packet from at least one network node, the first packet comprising an encoding vector; based on at least the encoding vector, determining a status of the first packet at the network node; using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and transmitting the plurality of encoded second packets to the at least one network node.

[0007] In an additional exemplary embodiment, a computer program product is disclosed comprising a computer-readable memory medium bearing computer program code embodied therein for use with a computer, the computer program code comprising: code for receiving a first packet from at least one network node, the first packet comprising an encoding vector; code for, based on at least the encoding vector, determining a status of the first packet at the network node; code for, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; code for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and code for transmitting the plurality of encoded second packets to the at least one network node.

[0008] In a further exemplary embodiment, an apparatus is disclosed that includes a means for receiving a first packet from at least one network node, the first packet comprising an encoding vector; means responsive to at least the encoding vector for determining a status of the first packet at the network node; means responsive to the determined status of the first packet, for making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; means for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and means for transmitting the plurality of encoded second packets to the at least one network node.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The foregoing and other aspects of embodiments of this invention are made more evident in the following Detailed Description of Exemplary Embodiments, when read in conjunction with the attached Drawing Figures, wherein:

[0010] FIG. 1 is a block diagram of a number of network nodes communicating in a mixed wired and wireless network;

[0011] FIG. 2 is a block diagram of a number of communications layers;

[0012] FIG. 3 is a graph that presents two probabilities that are functions of the size of the subset a , $|a|$, where a is in $[0, 232]$;

[0013] FIG. 4 is a graph that presents the density of all transmitted packets (packet 1 to 1463) for different NC approaches;

[0014] FIG. 5 is a graph that presents the total complexity for all encoded/recoded packets for different NC approaches;

[0015] FIG. 6 is a graph that presents the total complexity for all packets decoded at one sink for different NC approaches;

[0016] FIG. 7 is a block diagram of actions performed for encoding data in accordance with an exemplary embodiment of the invention;

[0017] FIG. 8 is a block diagram of actions performed for deciphering and saving information for received packets and associated network nodes in accordance with an exemplary embodiment of the invention; and

[0018] FIG. 9 is a block diagram of actions performed for decoding information in accordance with an exemplary embodiment of the invention.

DETAILED DESCRIPTION

[0019] Before proceeding with a description of the exemplary embodiments, an overview of network coding (NC) is helpful. Referring now to FIG. 1, a block diagram is shown of a number of network nodes 10 communicating in a mixed network 100. The mixed network 100 includes a wireless network 20 and a wired network 30. The wireless network 20 includes network nodes 10-1 and 10-2 in wireless communication. The wired network 30 includes network node 10-2 and 10-3 in wired communication. The network node 10-1 includes one or more processors 10-A1, one or more memories 10-B1 containing one or more programs 10-C1, a radio frequency transceiver 10-D1 able to receive and transmit data, a wired network interface 10-E1, and one or more antennas 10-F1. Similarly, the network node 10-2 includes one or more processors 10-A2, one or more memories 10-B2 containing one or more programs 10-C2, a radio frequency transceiver 10-D2 able to receive and transmit data, a wired network interface 10-E2, and one or more antennas 10-F2. In the example of FIG. 1, the network node 10-3 includes one or more processors 10-A3, one or more memories 10-B3 containing one or more programs 10-C3, and a wired network interface 10-E3. NC is applicable to wired-only, wireless-only or wired and wireless networks. For instance, the network nodes 10-1 and 10-2 could be mobile devices, such as cellular devices, and the network node 10-2 could be a base station. As another example, each of the network nodes 10 could be a computer in a wired network. FIG. 1 is merely illustrative of possible devices suitable for use with NC.

[0020] Turning to FIG. 2, as described above, the one or more programs 10-1 can be logically separated into a set 200 of communications layers. For instance, an application layer 210 is "above" a coding layer 220, which in this example implements NC techniques to produce network coded data (not shown in FIG. 2). Additional layers include the transport layer 230, the Internet layer 240, and the link/physical layer 250. The link/physical layer 250 produces data suitable for communicating over one or more wired or wireless channel(s) 270. It is noted that this figure is merely illustrative, as there will typically be more layers than what is shown here. The coding layer 220 is where NC takes place, in this example. It is also noted that the NC can, instead of residing in coding layer 220, be part of any layer between an application (e.g., interfacing with or part of application layer 210) and the

physical layer 230, and may not comprise an entire layer. The coding layer 220 may also be implemented as part of an application (e.g., as part of application layer 210).

[0021] It is noted that FIGS. 1 and 2 are both applicable to NC using and not using the techniques (described below) of the invention. In one exemplary embodiment, the program(s) 10-C are configured to use the techniques of the invention. In another exemplary embodiment, the techniques of the invention are implemented in hardware, such as an integrated circuit.

[0022] **Overview of NC**

[0023] To proceed with an overview of NC, in routing networks, data to be transferred from a source to a sink via the network is divided into n discrete blocks. Hence, the source creates n disjoint packets, where each contains a block of the original data and holds no information about the other $n-1$ packets. Packets are therefore numbered at the transmitter (i.e., part of the source) and must be organized accordingly at the receiver to ensure coherent reconstruction. For the sink to reconstruct the data, the sink must hold all n packets.

[0024] In a NC-system, a packet contains a linear representation of a subset of the original data or the entire original data. The values of the original bits are scattered across several packets and each packet can be thought of as a many \rightarrow 1 mapping of the original data. This representation of data is related to FEC which is frequently used in digital communication systems. Therefore, the source can construct $g+r$ linear combinations of the original data, where $r \in \mathbb{Z}^+ \gg g=n$, of which a subset of approximately size g describes the original data completely. Hence, if a sink possesses any g linear independent packets of the $g+r$ created packets from the source, the sink can reconstruct the original data.

[0025] Routing problems can be solved or simplified by using NC, as the requirement for decoding at the source is reduced from all n packets to any g packets. In some networks, NC can be used to increase the throughput above what is achievable with routing.

[0026] Security and NC is closely related, as all coded packets in a NC-system are encrypted by default. If exactly g packets are transmitted from the source to the sink, then an intruder must intercept all g packets to be able to decode any information.

[0027] In a point-to-point (P2P) application, NC can reduce the problem of locally rare parts, as the data can be maximally spread in the network. This is because in a NC system all packets are equally rare, namely unique, and new unique packets can be generated at nodes in the network.

[0028] Coding Operations in NC

[0029] Data to be transferred from the source to the sinks is divided into packets of length m . The number of original packets over which encoding is performed is typically referred to as the batch size or generation size and denoted g . Thus the g original data packets of length m are arranged in the matrix $M = [m_1 \ m_2 \ \dots \ m_g]$, where m_i is a column vector.

[0030] Encoding

[0031] To encode a packet x at the source, M is multiplied with a randomly generated vector g of length g , $x = M \times g$. In this way, one can construct $X = [x_1 \ x_2 \ \dots \ x_{g+r}]$ that consists of $g+r$ coded data packets and $G = [g_1 \ g_2 \ \dots \ g_{g+r}]$ that contains $g+r$ randomly generated encoding vectors, where r is the number of redundant packets.

[0032] Note that if an encoding vector consists of all zeros except a single scalar that is one, the coded packet is equal to an original packet and it is said that the coded packet is trivially encoded.

[0033] Recoding

[0034] Any relay or sink node that has received $g-i > l$ linear independent packets can recode and thus create new coded packets. All received coded packets are placed in the matrix $X^\wedge = [x^\wedge_1 \ x^\wedge_2 \ \dots \ x^\wedge_{g-i}]$ and all encoding vectors are placed in the matrix $G^\wedge = [g^\wedge_1 \ g^\wedge_2 \ \dots \ g^\wedge_{g-i}]$, and this latter matrix is denoted as the decoding matrix. The number of received linear independent packets $g-i$ is equal to the rank of G^\wedge . G^\wedge and X^\wedge is multiplied with a randomly generated vector h of length $g-i$, $g^\sim = G^\wedge \times h$, $x^\sim = X^\wedge \times h$. In this way, one can construct $G^\sim = [g^\sim_1 \ g^\sim_2 \ \dots \ g^\sim_{g-i}]$ that contains $g-i$ randomly generated recoding vectors and $X^\sim = [x^\sim_1 \ x^\sim_2 \ \dots \ x^\sim_{g-i}]$ that consists of $g-i$ recoded data packets.

[0035] Note that h is only used locally and that there is no need to distinguish between coded and recoded packets when further recoding or decoding is done.

[0036] Decoding

[0037] In order for a sink to successfully decode the original data packets, it must receive g linear independent coded packets and encoding vectors, thus the rank of the decoding matrix is g . All received coded packets are placed in the matrix $X = [x_1 \ x_2 \ \dots \ x_g]$ and all encoding vectors are placed in the matrix $G = [g_1 \ g_2 \ \dots \ g_g]$. The original data M can then be decoded as $M = X \times G^{-1}$.

[0038] **Density**

[0039] The density, d , sometimes referred to as the degree, of a coded packet is defined by its encoding vector, g , and states how big a ratio of the original packet, the coded packet is a combination of:

$$d(g) = (\sum_{i=1}^g \{g_i > 0\}) / g \quad (1)$$

[0040] Thus if q (the field size) is reasonably high, most scalars in a random generated encoding vector will be non-zero and $d \approx 1$. For a small field, a higher ratio of scalars in a random generated encoding vector will be zero, e.g. for the binary field, $d = 1/2$.

[0041] **Coding Complexity**

[0042] The O-notation complexity of the coding operations, per packet, is the following:

$$\text{Encoding: } O(g(r+m)) \sim O(g \cdot m); \quad (2)$$

$$\text{Recoding: } O((g-z)(r+m + g-i)) \sim O((g-i) \cdot m + (g-i)^2); \quad (3)$$

$$\text{Decoding: } O(g \cdot m + g^2). \quad (4)$$

[0043] See Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, and Torben Larsen, "Cautious view on network coding - from theory to practice", Journal of Communications and Networks (JCN), 10(4):403-411, December 2008; and C. Fragouli, J. Boudec, and J. Widmer, "Network coding: an instant primer", SIGCOMM Comput. Commun. Rev., vol. 36, no. 1, pp. 63-68, 2006. The cost of generating a random number r is system specific and here set to zero.

[0044] **Exemplary Problems with Conventional NC**

[0045] Today NC is often suggested to be implemented with random linear codes coded over a Galois Field of size q , where q is some relatively large number e.g.

$\{2^{16}, 2^{32}\}$. See T. Ho, R. Koetter, M. Medard, D. Karger, and M. Ros, "The benefits of coding over routing in a randomized setting", in Proc. IEEE ISIT, 2003. This solution provides very low expected overhead in terms of expected retransmissions, due to a lower probability of generating linear dependent packets, but it is non-trivial to construct an implementation with high coding throughput. See Janus Heide et al., cited above. It is noted that a Galois field is also called a finite field and is a field that contains a finite number of elements.

[0046] Systematic coding reduces the density of the first g packets created at the source to $1/g$. However, all subsequently coded packets generated at the source and all coded packets generated at a relay still have density 1 .

[0047] With regard to NC, all coded packets constructed at a node are linear combinations of all packets that node has received when coding the new packet. Thus, as a node receives more linear combinations, the node will become a better source, but the computations needed to create a coded packet will also increase. That is, coded packets constructed at the source are a combination of all g packets, as the source possesses all g packets, and thus all g elements in g will be non-zero. When a coded packet is received and decoded at a node, the number of operations that are needed to decode this packet is determined by the ratio of non-zero elements in g , often referred to as the density (as described above).

[0048] Thus, if the density is reduced, a packet can be decoded with fewer operations, which can increase the decoding throughput. However, if the density is reduced randomly, e.g., by setting random elements to zero, the probability that the packet is linear dependent at the receiving node increases.

[0049] **Exemplary Embodiments**

[0050] Certain exemplary embodiments herein solve the above problems by intelligently reducing the density of coding vectors at sources and relays, based on intrinsic information embedded into the encoding vector of each coded packet.

[0051] This can be done by splitting the used field over which packets are coded into two or more subsets. Moreover, each scalar in an encoding vector will thus belong to one of these subsets and consequently indicate the status of that corresponding packet at the sender of the packet, such as: not received, received but not decoded, decoded, etc. This differs from current NC systems, where the scalars in

the encoding hold no information about the status of packets at the sender of the packet.

[0052] Below, the exemplary embodiments of the invention are considered implemented in software and thus are described as such. For example, the software could be implemented as one or more programs 10-C in FIG. 1 residing in one or more memories 10-B. The one or more programs 10-C comprise program code, and the one or more memories 10-B and the program code are configured to, with the one or more processors 10-A, cause a network node 10 to perform actions described below. The one or more programs 10-C can be implemented as shown in FIG. 2, where the coding layer 220 performs the coding (include decoding) operations described below. It is noted that each network node 10 in FIG. 1 can be a source, relay, or sink. The exemplary embodiments may also be implemented as computer program product comprising a computer-readable memory medium (such as a compact disk, digital versatile disk, or one or more memories 10-B) bearing computer program code embodied therein for use with a computer. However, the invention may be implemented in hardware, such as an integrated circuit or programmable logic circuit (i.e., IC/PLC 70) or some combination thereof.

[0053] It is assumed that coding operations are performed over a Galois Field, specifically $q=2^{32}$ is used in the following evaluation. Note that smaller values of q can also be used, and that $q=\{2^8, 2^{16}, 2^{32}\}$ are good candidates as the corresponding data containers are directly supported in most computer systems.

[0054] Field subsets

[0055] The used Galois field of size q is divided into s sets a_1, a_2, \dots, a_s , where $|a_1| + |a_2| + \dots + |a_s| = q$. Here, $n=2$ is considered and put the elements $\{0, 1, 2, \dots, a\}$ values into set a and the remaining elements $\{a+1, a+2, \dots, q\}$ into a' . This is practical as for any scalar b , where $b < a$, it holds that $b+1 \leq a$. In general for any scalar b , where $a_i < b < a_{i+1}$, it holds that $a_i \leq b+1 \leq a_{i+1}$, which will be utilized later (see "Second approach, below).

[0056] Encoding

[0057] When encoding, all scalars can be chosen arbitrarily because the source (e.g., a network node 10 in FIG. 1) holds all original packets and thus all possible g are valid encoding vectors.

[0058] Based on information collected from received coded packets (see decoding), the encoding node can determine for which packets all (or certain) nodes have a pivot element. A pivot element is the element of a finite set, which is selected first by an algorithm to perform certain calculations. Each node has between zero and g pivot elements for each generation. If a node has zero, the node has received no information. If the node has g , the node has received all the data. Whenever a node receives one packet, the node will get one additional pivot element for that generation. This is under the condition that the received packet is independent of all previously packets. When constructing a coded packet, these packets can be omitted from the linear combination in order to reduce the number of operations. That is, when the encoding node determines that a receiving node, to be sent an encoded packet corresponding to a received packet from the receiving node, has a pivot element corresponding to the received packet, the encoding node would not use the received packet in the generation of the linear combination of packets to be sent. This means the received packet would not be sent to the receiving node. This process can be generalized to all nodes to which the encoding node can communicate.

[0059] **Recoding**

[0060] When a node (e.g., a network node 10 of FIG. 1) recodes, g is not directly specified but is calculated based on the generated h and the received encoding vectors G^{\wedge} , thus not all elements in the resulting g can be chosen arbitrarily. Only the g - i scalars, for which the node has pivot elements in the decoding matrix, can be chosen arbitrarily. This can be done by choosing the scalars in h within a specific range for each h_i that corresponds to a row where the node has a pivot element. One can denote the set of g - i scalars in \mathbf{A} for which the node have pivot elements in the corresponding rows, h' . Thus, if all scalars in h are set to some value c , all elements in h' will be equal to c and the remaining elements will be values in q .

[0061] The reason is that for the columns corresponding to h' only one row will contain a one and all other rows a zero. Thus, when all rows are combined to form a coded packet, the scalars that correspond to packets in h' element can be chosen arbitrarily by multiplying the corresponding row with a scalar.

[0062] Based on the information collected from received coded packets (see decoding below), a node can determine for which packets all (or certain) nodes

have a pivot element. When recoding a packet, these packets can be omitted from the linear combination.

[0063] Decoding

[0064] When receiving a coded packet, a node must inspect the encoding vector, decipher and save the information embedded in the encoding vector.

Therefore, all nodes must use the same rules to create and decipher encoding vectors, this can be predefined or appended to the encoding vector.

[0065] As the node may receive data from several sources, the node should save this information separately for each source. If the node previously received a coded packet from the same node, the receiving node should update the corresponding information, e.g., in the one or more memories 10-B. The corresponding information includes information about whether the sending node had a pivot element for the given packet. The corresponding information may also include into which sets s the coded packet falls.

[0066] First approach

[0067] One very simple approach is to let a specific value, e.g., the value one, in the encoding vector denote that the coding node have a pivot element for the corresponding packet. Thus when a node encodes, g will be a vector of ones of length g , because the node holds all original data.

[0068] When recoding, the recoding vector h is constructed by setting each of the values in h' to one and drawing the rest of the values randomly from $[0,q]/\{1\}$. Then the new encoding vector g_{\sim} is calculated by $g_{\sim} = G^{\wedge} \times h$. All resulting values in g_{\sim} that corresponds to h' will then be one. If any values in g_{\sim} not corresponding to h' are equal to one, the chosen h is not usable, therefore it is discarded and a new h is generated. The data is then encoded using the generated h , $x_{\sim} = X^{\wedge} \times h$.

[0069] Thus, when a node n_1 receives a packet from node n_2 , the node n_1 inspects the encoding vector. For each value in the received g that is equal to one, node n_1 knows that node n_2 has a pivot element. If node n_1 then generates a reply in form of a coded packet, node n_1 can set all these values to zero in its encoding vector in order to reduce the number of operations, without decreasing the probability that the packet will be useful at node n_2 . However in order to do this, node n_1 must also have pivot elements for the same packets. Thus node n_1 can set all values in g to zero for the common set of h' in node n_1 and h' in node n_2 .

[0070] As a side node, this approach is not particularly practical, as it uses only a single scalar to indicate a pivot element. This is particularly obvious when encoding, because in this case the node have pivot elements for all packets and thus there is only one valid encoding vector namely all ones. However, it is possible to use this approach.

[0071] **Second approach**

[0072] To improve the first approach, one can increase the size of the subset $|a|$ that indicates that the sender has a pivot element. One can choose $a = [0, a]$, thus $|a| = a + 1$, and for any scalar b , where $b < a$, it holds that $b+1 \leq a < |a|$. To identify a good value of a , it is necessary to determine the probabilities that influence the performance of a system where this approach is implemented. Namely, the probability that the resulting values in g are in a for packets that are not in h' , this is denoted $p_{i, \alpha}$, and the probability that an encoding vector g is linear dependent at the receiving node, this is denoted $p_{\text{dependent}}$. These probabilities should be as small as possible, as these represent outcomes that will introduce (that is, increase) costs.

[0073] When some value in the resulting g is in a for some packet not in A' , a new h must be generated. Thus, the node will perform additional operations in order to encode a packet, which is what this invention seeks to reduce.

[0074] The probability that a random resulting scalar in g is in a is the following:

$$|a|/q$$

The probability that all $g-i$ scalars are not in a is thus:

$$(1 - |a|/q)^{g-i}$$

And hence the probability that any of the $g-i$ scalars are in a is the following:

$$p_{\text{in } a} = 1 - (1 - |a|/q)^{g-i} \quad (5)$$

[0075] When a node receives a coded packet that is linearly dependent with the packets the node already has, the packet is not useful at that node and thus the

linearly dependent coded packet introduces wasted resources. Additionally, it is necessary to perform operations on the packet to determine that it is linearly dependent, and these operations will also be wasted.

[0076] The probability that a packet is linearly dependent at the receiving nodes is a function of the field size and the rank deficiency at that node. See Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, and Torben Larsen, "Network coding for mobile devices - systematic binary random rateless codes", in The IEEE International Conference on Communications (ICC), Dresden, Germany, 14-18 June 2009. The field over which coding operations is performed is effectively reduced for the packet for which the coding nodes have pivot elements. Thus in the case where a node n_1 is the sending node and has rank g , and the receiving node n_2 has rank $g-i$, the probability that a received packet at n_2 is linear dependent is.

$$P_{\text{dependent}} = 1/|F|^{i-1} \quad (6)$$

[0077] If the sending node n_1 has rank $g-i$, where $i > 0$, this probability is less straightforward and depends on the correlation between the received packets at the sending node n_1 and the receiving node n_2 .

[0078] FIG. 3 presents two probabilities that are functions of the size of the subset a , $|a|$, where a is in $[0, 2^{32}]$. In this example, $q=2^{32}$ and $g=1024$. The probabilities also depend on the rank deficiency at the sink, i , here the worst case values are used, for $p_{i,\alpha}$ $i=g$, and for $P_{\text{dependent}}$ $i=1$. As described above, $p_{i,\alpha}$ is the probability that, when a node generates an encoding or recoding vector, at least one of the elements in the encoding vector is in a . Thus, a new encoding or recoding vector must be generated at the node. As also described above, $P_{\text{dependent}}$ is the probability that a received coded packet is linearly dependent at the sink. Thus a new packet must be generated and sent from a source. The plot shows as $|a|$ grows, $p_{i,\alpha}$ grows and $P_{\text{dependent}}$ decreases. For $|a|$ equal to one, $p_{i,\alpha}$ equals one, for $|a| > 10$ $P_{\text{dependent}}$ equals one.

[0079] Consequently, FIG. 3 shows the trade off that exists between $p_{i,\alpha}$ and $P_{\text{dependent}}$ as a function of $|a|$. Both the occurrences that these probabilities correspond to are associated with some cost. The cost for $P_{\text{dependent}}$ is largest, but the ratio of the cost between the two occurrences is system specific and thus there exist no

general best choice of $|a|$. In the following, it is assumed that $|a|$ is chosen so that $p_{\text{dependent}}$ is so close to zero that $p_{\text{dependent}}$ can be ignored.

[0080] Packet Density

[0081] A source holds some data $M_{m \times g}$ that is to be transmitted to N sinks r_1, r_2, \dots, r_N . All nodes are connected via one broadcast channel and erasures occur on this channel with some probability P . If g is relatively big, each sink will have received g packets when $g \cdot (1-P)^{-1}$ packets have been transmitted on the channel. These packets can be transmitted by the source or the sinks, however initially only the source may transmit as it is the only one that holds the data.

[0082] With traditional network coding, each packet (denoted j), all packets are coded packets and are combinations of all g original packets and thus $d(j) = \setminus$.

$$d(j) = \setminus \quad \text{for } j \text{ in } \{1, 2, \dots, g \cdot (1-P)^{-1}\}. \quad (7)$$

In systematic network coding, the first g packets are uncoded, and thus are a combination of a single packet. The remaining packets are linear combinations of all g original packets. Consequently:

$$d(j) = \setminus g \quad \text{for; in } \{1, 2, \dots, g\}; \text{ and} \quad (8)$$

$$d(j) = 1 \quad \text{for; in } \{g+1, g+2, \dots, g \cdot (1-P)^{-1}\}. \quad (9)$$

[0083] The suggested approach in the instant invention is also systematic and thus the first g packets are uncoded. The following coded packet is a linear combination of the packet that not all nodes have pivot elements for. Packet losses at the nodes are assumed to be uncorrelated, which also presents the worst case scenario. Thus when packet j is to be transmitted, all sinks have received $(1-P)^{-(j-1)}$ packets. Therefore, the probability that a sink has a pivot element for a packet is $(1-P)^{-(j-1)}/g$. The probability that all sinks have a pivot element for a packet is the probability that one sink has the pivot element, raised to the number of sinks:

$$d(j) = \setminus g \quad \text{for; in } \{1, 2, \dots, g\} \quad (10)$$

$$p_{\text{packet}}(j) = (1-P)^{-(j-1)}/g$$

$$\text{for } j \text{ in } \{g+1, g+2, \dots, g \cdot (1-P)^{-1}\}$$

$$d(j) = 1 - P_{\text{packet}}(j)^n \quad \text{for } j \text{ in } \{g+1, g+2, \dots, g * (-P)^{-1}\}. \quad (11)$$

[0084] The resulting densities of encoded packets for $g=1024$, $P=0.3$, and N in $\{1, 2, 4, 8\}$ are plotted on FIG. 4. As can be seen, the suggested approach herein reduces the density of the coded packets, especially when the number of receiving nodes is low. More particularly, FIG. 4 presents the density of all transmitted packets (packet 1 to 1463) for different NC approaches. With regard to traditional NC and systematic NC, the performance does not depend on the number of receivers, N , but the new approach does and is plotted for 1, 2, 4 or 8 receivers. In this example the packet error probability (PEP) is 0.3 and $g = 1024$.

[0085] For traditional NC, the density is constant and one. For systematic NC, the density is low until $i=g$, after which the density increases to one. For the suggested approach herein, the density is low until $i=g$, after which the density jumps and then decreases towards zero. The density from the g th packet depends on the number of sinks, N , and when N increases, the density increases.

[0086] System Complexity

[0087] To calculate the resulting complexities, expressions are needed for the number of packets from which each coded packet is a linear combination. This is denoted as g' :

$$g'(j) = g * d(j). \quad (12)$$

[0088] The resulting complexities can then be calculated from the complexity expressions where g is substituted for encoding and recoding with g' , and $g-i$ with g' for recoding. In wireless networks, packets are typically approximately 1500 bytes in size, as $q=2^{32}$ the symbol size is 32 bits, and thus packet length, m , is 375.

$$\text{Encoding:} \quad O(g' * m) \quad (13)$$

$$\text{Recoding and decoding:} \quad O(g' * m + g'^2) \quad (14)$$

To include the PEP at the sinks, the probability that each packet is received ($1-P$) is multiplied to the cost for each packet, in terms of complexity, $(1-P) * O(g' * m + g'^2)$.

[0089] As described above, FIG. 4 shows the density of all coded packets, for different approaches. When the new approach presented herein is used, the density of all packets is reduced or the same as the systematic approach and the new approach performs better than the traditional approach. When the number of receiving nodes is low, the new approach performs best, and in the worst case where N is infinite, the performance is identically to that of the systematic approach, although this is not depicted on the plot.

[0090] The number of operations performed when different approaches are used, and for different P is plotted in FIGS. 5 and 6. FIG. 5 shows operations for encoding, and FIG. 6 shows operations for decoding. FIG. 5 presents the total complexity for all encoded/recoded packets for different NC approaches. With regard to traditional NC and systematic NC, the performance does not depend on the number of receivers, N , but the new approach does and is plotted for 1, 2, 4 or 8 receivers. P is varied between 0 and 0.5, $g = 1024$, and $m=3T5$. The plotted values are the sum of the complexity expressions for all encoded/recoded packets that are generated. As P increases, the number of performed operations increases for all approaches. The new approach requires the least amount of operations and performs best when the number of receiving nodes is low.

[0091] FIG. 6 presents the total complexity for all packets decoded at one sink for different NC approaches. Regarding traditional NC and systematic NC, the performance does not depend on the number of receivers, N , but the new approach does and is plotted for 1, 2, 4 or 8 receivers. P is varied between 0 and 0.5, $g = 1024$, and $w=375$. The plotted values are the sum of the complexity expressions for all packets decoded at a sink. As P increases, the number of performed operations increases for all approaches. The new approach requires the fewest number of operations and performs best when the number of receiving nodes is low. Both show a significant reduction in the number of performed operations with the suggested approach. This can translate into higher coding throughput, lower central processing unit (CPU) utilization and reduced power usage.

[0092] An exemplary proposed technique is focused on meshed networking with device-to-device communication (exchanging any kind of information), as the technique reduces the amount of signaling overhead and assures efficient coding. For instance, the technique can be used for cooperative exchanges

among devices of downloaded information. In particular, the technique is useful in scenarios where the complexity of standard coding could potentially limit the throughput, e.g., if the transmitting nodes have low computational capabilities or if the data rate is high.

[0093] FIG. 7 is a block diagram 700 of actions performed for encoding data in accordance with an exemplary embodiment of the invention. The block diagram (and the block diagrams shown in FIGS. 8 and 9) may be a method performed by an apparatus such as network node 10 (see FIG. 1), e.g., under control of one or more processors 10-A. The block diagram 700 (and the block diagrams shown in FIGS. 8 and 9) may be a sequence of actions carried out by hardware, such as an integrated circuit, programmable logic device, or other devices, or combinations of these. The actions in block diagram 700 (and the block diagrams shown in FIGS. 8 and 9) may also be a computer program product comprising a computer-readable memory medium (e.g., one or more memories 10-C in FIG. 1, a compact disk, a digital versatile disk/digital video disk, or the like) bearing computer program code embodied therein for use with a computer (e.g., such as network node 10 of FIG. 1). The computer program code has code causing the computer to perform the actions shown in FIG. 7 (and the block diagrams shown in FIGS. 8 and 9).

[0094] It is helpful to note that packets are usually not retransmitted by a single network node. In other words, a source node might transmit X packets. Instead of retransmitting any of those X packets (e.g., if another node does not receive one or more of the X packets), a network node (such as a recoding node) that receives those X packets and would like to retransmit that data in the those X packet will encode (or recode) those X packets and retransmit them in the encoded/recoded state. It is also noted that the terms "encoded" and "coded" are used interchangeably herein.

[0095] In block diagram 700, encoding and recoding are both described, as recoding is a version of encoding. At this point, it is helpful to discuss the various options for network nodes (such as a network node 10 of FIG. 1). If the network node is a source and this is the first time data has been encoded and transmitted from the source to other network nodes, uncoded data 730 will be used in block 731 and the first g packets are transmitted (to other network nodes 10) (block 731). If the network node is only a source (and not a sink or a recoding node), the source only performs block 730 and 731. Multiple sources and recoding nodes will perform the additional

blocks in block diagram 700. It is noted that some header information is added to the packet (such as packets 705 or 765) for the receiver to know if the data in the packet is uncoded or coded and how the data is coded. In the case of the uncoded packet, this information can be compressed a lot. This header information is outside the scope of this disclosure and is known to those skilled in the art.

[0096] For nodes performing coding, the initially received packets (in this example) will be packets of uncoded data 730. Block 715 will not be performed until coded packets are received, so a secondary source or recoding node at this point does not perform block 715. After a certain number of uncoded packets are received, the network node forms linear combinations of encoded packets using, in this case, the uncoded received packets (block 720). For block 735, any scalar from the finite field may be used for encoding (block 745). Block 740 is not performed at this time. Block 720 produces encoded packets 765, which include coded data 766 and an encoding vector 767 (as described above). These encoded packets 765 are transmitted in block 725. At this time, blocks 750 and 755 are not performed.

[0097] Subsequently, the network node begins to receive encoded packets 705 from other nodes (block 710). Each received packet 705 includes coded data 706 and an encoding vector 707. It is noted that the received packets 705 are encoded, but will be referred to as "received packets" to distinguish from encoded packets 765 produced in block 720.

[0098] In block 715, the received packets are examined (more specifically, the encoding vector 707 is examined) and the received packets are deciphered and information is saved corresponding to the received packets and the nodes. This is explained in more detail in reference to FIG. 8.

[0099] After a certain number of encoded packets 705 are received, block 720 would be performed. In block 720, a linear combination of encoded packets is formed, where the secondary source/recoding network node performs block 720 by performing block 735 using encoded packets 705, and by determining if the information determined in block 715 indicates that other network nodes have the coded packets (or in an exemplary embodiment, the network nodes have pivot elements corresponding to the coded packets). Note that block 735 is subject to determining into which set of the sets of the finite field the scalars should fall, as described above in "Field subsets", "Encoding", "First approach", and "Second

approach". Furthermore, once encoded packets 705 are received, the secondary source/recoding network node begins recoding and performs block 750 instead of block 745. In block 750, as described above in "Recoding", only the $g-i$ scalars for which the recoding node has pivot elements in the decoding matrix can be chosen arbitrarily. The coded packets for which the information determined in block 715 indicates that other network nodes have the coded packets (or in an exemplary embodiment, the network nodes have pivot elements corresponding to the coded packets) are not included in the linear combination (block 740). What this means (block 755) is that these coded packets are not transmitted in block 725. In block 725, the remaining encoded packets are transmitted.

[00100] Turning to FIG. 8, a block diagram 800 is shown of actions performed for deciphering and saving information for received packets and associated network nodes in accordance with an exemplary embodiment of the invention. Block diagram 800 is one exemplary technique for performing block 715 of FIG. 7 (and also block 915 of FIG. 9). In block 805, it is determined for each encoding vector into which set of the sets s of the finite field (see "Field subsets" above) the encoding vector falls. In block 810, status of the received packet is determined based on the sets. Such status can be that the sending node has pivot nodes for the received packets (block 820) or can be other status (block 830). Such other status could be, as examples, not received 835, received but not decoded 840, or decoded 845.

[00101] In block 815, the status for received packets is updated for the packets and for the sending node. As an example, a first table 850 may be kept, where the table 850 includes entries 855-1 through 855-x corresponding to Node 1, and entries 860-1 through 860-y corresponding to Node 2. Entries for additional nodes are not shown. Each entry includes, in this example, an indication 865 of the node, and indication of the packet 870, and an indication 875 of the status. The table 850 is merely one example and is not to be considered limiting. For instance, smaller sets of data could be used, such as that shown in table 880, which contains entries 881-1 and 881-2 (other entries are not shown). Each entry has an indication 885 of a packet and an indication 890 of whether all other nodes have a pivot element corresponding to the packet or not.

[00102] Turning to FIG. 9, a block diagram 900 is shown of actions performed for decoding information in accordance with an exemplary embodiment of

the invention. Block diagram 900 would be performed by a sink. It is noted that some sinks may also be sources or recoding nodes, and the functions performed by those nodes may be performed as described above. Block diagram 900 begins in block 910, when packets are received, e.g., from source or recoding nodes. In block 915, the received packets are deciphered and information for the received packets and associated nodes is saved (e.g., as described above in block diagram 800 of FIG. 8). In block 925, matrices of coded data (e.g., 766) and encoding vectors (e.g., 767) are formed. In block 930, the matrices are used to decode the coded data. See the "Decoding" sections above. In block 935, the decoded packets are output. The block diagram 900 ends in block 950.

[00103] Embodiments of the present invention may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. The software, application logic and/or hardware may reside on any type of wired or wireless network node, such as nodes in a cellular network or nodes in a wired peer-to-peer or peers-to-servers network. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a "computer-readable memory medium" may be any memory media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer, with one example of a computer described and depicted in FIG. 1.

[00104] The embodiments of this invention may be implemented by computer software executable by one or more processors 10-A or by hardware. Further in this regard, it should be noted that the various blocks of the logic flow diagrams of FIGS. 7, 8, and 9 may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions.

[00105] The one or more memories 10-B may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The one or more processors 10-A may be of any type suitable to the local technical environment, and may include one or more of general purpose computers,

special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on a multi core processor architecture, as non limiting examples.

[00106] Embodiments of the inventions may be practiced in various components such as integrated circuit modules (e.g., IC 70). The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

[00107] Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

[00108] If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described functions may be optional or may be combined.

[00109] One exemplary technical effect of the invention is that it enables a reduced number of operations relative to other NC techniques. Another exemplary technical effect of the invention is that it enables reduced bandwidth, as certain packets will not be communicated when these packets are determined to exist at certain nodes in the network.

[00110] In an exemplary embodiment, a computer program product is disclosed comprising a computer-readable memory medium bearing computer program code embodied therein for use with a computer, the computer program code comprising: code for receiving a first packet from at least one network node, the first packet comprising an encoding vector; code for, based on at least the encoding vector, determining a status of the first packet at the network node; code for, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; code for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets

includes or does not include the first packet; and code for transmitting the plurality of encoded second packets to the at least one network node.

[00111] It is noted that any of the operations previously described may be implemented through means. For example, an apparatus may include a means for receiving a first packet from at least one network node, the first packet comprising an encoding vector; means responsive to at least the encoding vector for determining a status of the first packet at the network node; means responsive to the determined status of the first packet, for making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets; means for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and means for transmitting the plurality of encoded second packets to the at least one network node.

[00112] Although various aspects of the invention are set out in the independent claims, other aspects of the invention comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims.

[00113] It is also noted herein that while the above describes example embodiments of the invention, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications which may be made without departing from the scope of the present invention as defined in the appended claims.

CLAIMS

What is claimed is:

1. A method comprising:
receiving a first packet from at least one network node, the first packet comprising an encoding vector;
based on at least the encoding vector, determining a status of the first packet at the network node;
using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets;
encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and
transmitting the plurality of encoded second packets to the at least one network node.
2. The method of claim 1, wherein determining a status of the first packet further comprises determining into which set from a plurality of sets of a finite field the encoding vector falls.
3. The method of claim 2, wherein determining a status of the first packet further comprises determining, based on into which set the encoding vector falls, the status indicates the at least one network node has a pivot element corresponding to the first packet.
4. The method of claim 2, wherein determining a status of the first packet further comprises determining, based on into which set the encoding vector falls, the status indicates the at least one network node has one of the following: not received the first packet, received but not decoded the first packet, or decoded the first packet.

5. The method of claim 1, wherein the at least one network node comprises a plurality of network nodes, and wherein making a decision further comprises making the decision that a first packet should be used as a packet in the encoding of the plurality of second packets only when the status of that packet for each of the plurality of nodes is a predetermined status.

6. The method of claim 2, wherein all possible vectors from the finite field are able to be used in the encoding and the encoding further comprises, for each packet of the plurality of packets, determining into which set the packet should be placed and placing the packet into that set by selecting one of one or more possible vectors in the set.

7. The method of claim 3, wherein the method is performed by a node, wherein there are g possible second packets, wherein each of the first packets also comprises coded data, wherein receiving further comprises receiving i first packets, and wherein encoding further comprises:

forming a matrix, $X^A = [x^A_1 \ x^A_2 \ \dots \ x^A_g]$, wherein each x^A is coded data from a respective one of the i first packets;

forming a decoding matrix $G^A = [g^A_1 \ g^A_2 \ \dots \ g^A_{g-i}]$, wherein each g^A is an encoding vector from a respective one of the i first packets;

generating a matrix h comprising a plurality of randomly selected scalars h_j , the generating including choosing the scalars in h within a specific range for each h_j that corresponds to a row in X^A where the node has a pivot element; and

multiplying the matrix, X^A and the generated matrix h to create the plurality of second packets.

8. The method of any one of the preceding claims wherein encoding further comprises forming the plurality of coded second packets as a number of linear combinations of data elements in the second packets.

9. The method of claim 8, wherein the data elements comprise one of uncoded data or coded data.
10. The method of any one of the preceding claims wherein each of the first packets comprises coded data and each of the encoded second packets comprises coded data and an encoding vector.
11. The method of claim 1, wherein receiving further comprises receiving a plurality of first packets, each first packet also comprising coded data, wherein the method further comprises forming a first matrix comprising the coded data, forming a second matrix comprising the encoding vectors, decoding the coded data using the first and second matrices, and outputting decoded data.
12. A computer program, comprising:
code for receiving a first packet from at least one network node, the first packet comprising an encoding vector;
code for, based on at least the encoding vector, determining a status of the first packet at the network node;
code for, using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets;
code for encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and
code for causing transmission of the plurality of encoded second packets to the at least one network node;
when the computer program is run on a processor.
13. The computer program according to claim 12, wherein the computer program is a computer program product comprising a computer-readable medium bearing computer program code embodied therein for use with a computer.

14. An apparatus, comprising:
 - at least one processor; and
 - at least one memory including computer program code,the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:
 - receiving a first packet from at least one network node, the first packet comprising an encoding vector;
 - based on at least the encoding vector, determining a status of the first packet at the network node;
 - using the determined status of the first packet, making a decision whether the first packet should be used or should not be used as a packet in an encoding of a plurality of second packets;
 - encoding the plurality of second packets to create a plurality of encoded second packets, wherein based on the decision the plurality of second packets includes or does not include the first packet; and
 - transmitting the plurality of encoded second packets to the at least one network node.

15. The apparatus of claim 14, wherein determining a status of the first packet further comprises determining into which set from a plurality of sets of a finite field the encoding vector falls.

16. The apparatus of claim 15, wherein determining a status of the first packet further comprises determining, based on into which set the encoding vector falls, the status indicates the at least one network node has a pivot element corresponding to the first packet.

17. The apparatus of claim 15, wherein determining a status of the first packet further comprises determining, based on into which set the encoding vector falls, the status indicates the at least one network node has one of the following: not received the first packet, received but not decoded the first packet, or decoded the first packet.

18. The apparatus of claim 14, wherein the at least one network node comprises a plurality of network nodes, and wherein making a decision further comprises making the decision that a first packet should be used as a packet in the encoding of the plurality of second packets only when the status of that packet for each of the plurality of nodes is a predetermined status.

19. The apparatus of claim 15, wherein all possible vectors from the finite field are able to be used in the encoding and the encoding further comprises, for each packet of the plurality of packets, determining into which set the packet should be placed and placing the packet into that set by selecting one of one or more possible vectors in the set.

20. The apparatus of claim 16, wherein there are g possible second packets, wherein each of the first packets also comprises coded data, wherein receiving further comprises receiving i first packets, and wherein encoding further comprises:

forming a matrix, $X^\wedge = [\chi^\wedge_1 \ x^\wedge_2 \ \dots \ x^\wedge_{g \cdot i}]$, wherein each χ^\wedge is coded data from a respective one of the i first packets;

forming a decoding matrix $C^\wedge = [g^\wedge_1 \ g^\wedge_2 \ \dots \ g^\wedge_i]$, wherein each g^\wedge is an encoding vector from a respective one of the i first packets;

generating a matrix h comprising a plurality of randomly selected scalars h_{ij} , the generating including choosing the scalars in h within a specific range for each h_{ij} that corresponds to a row in X^\wedge where the apparatus has a pivot element; and

multiplying the matrix, X^\wedge and the generated matrix h to create the plurality of second packets.

21. The apparatus of any one of claims 14 to 20, wherein encoding further comprises forming the plurality of coded second packets as a number of linear combinations of data elements in the second packets.
22. The apparatus of claim 21, wherein the data elements comprise one of uncoded data or coded data.
23. The apparatus of any one of claims 14 to 22, wherein each of the first packets comprises coded data and each of the encoded second packets comprises coded data and an encoding vector.
24. The apparatus of claim 14, wherein receiving further comprises receiving a plurality of first packets, each first packet also comprising coded data, wherein the apparatus further comprises forming a first matrix comprising the coded data, forming a second matrix comprising the encoding vectors, decoding the coded data using the first and second matrices, and outputting decoded data.

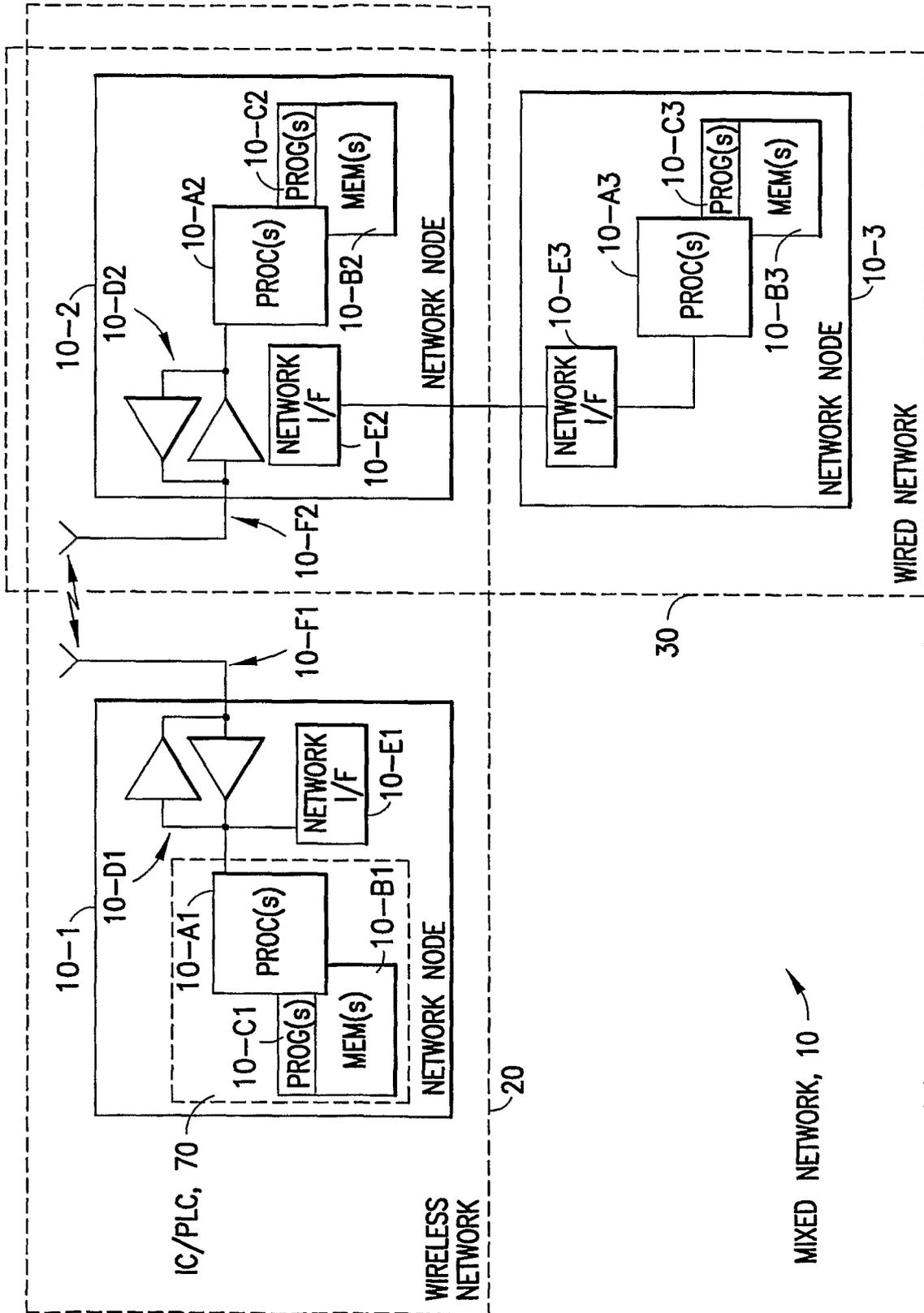


FIG. 1

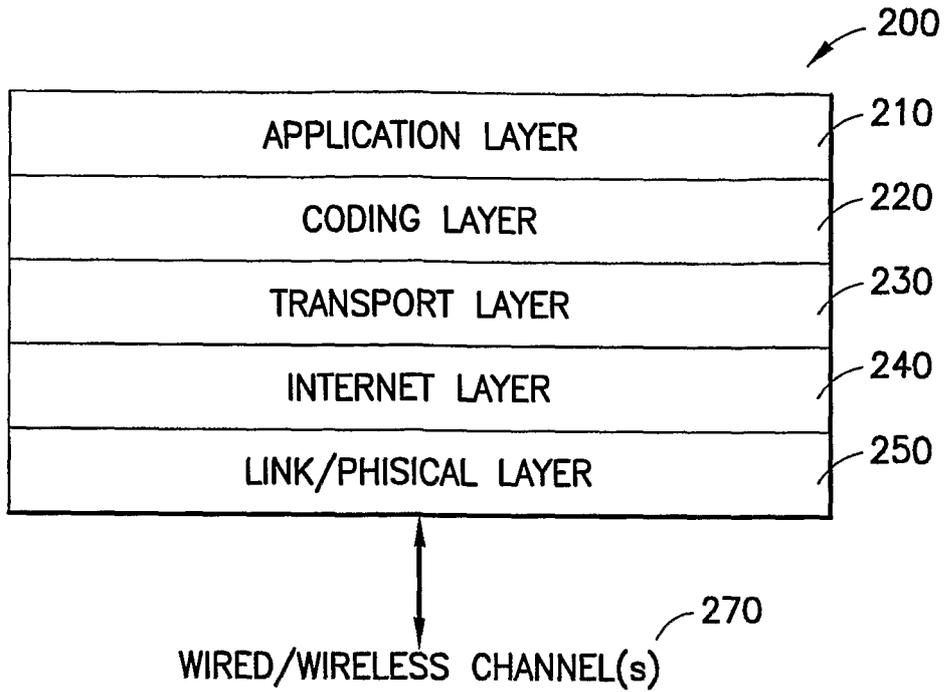


FIG.2

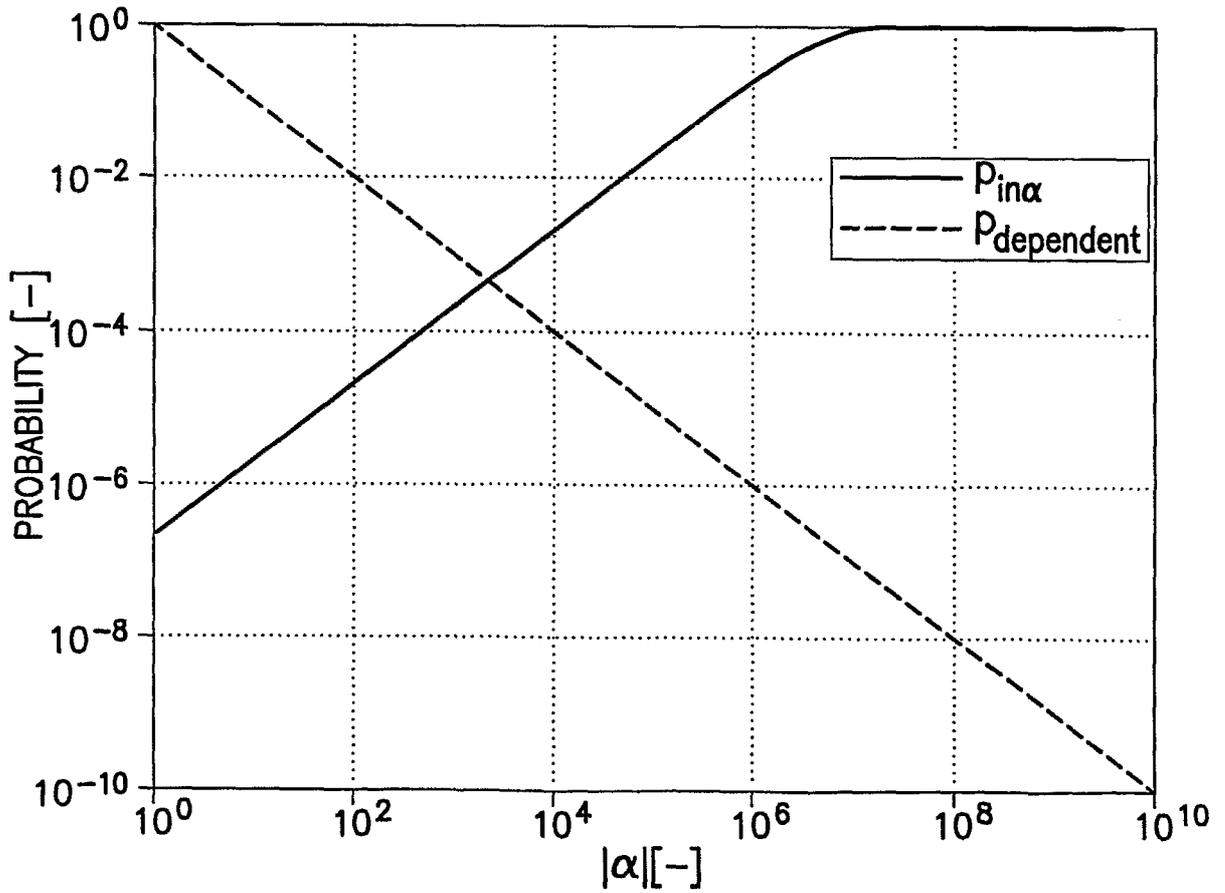


FIG.3

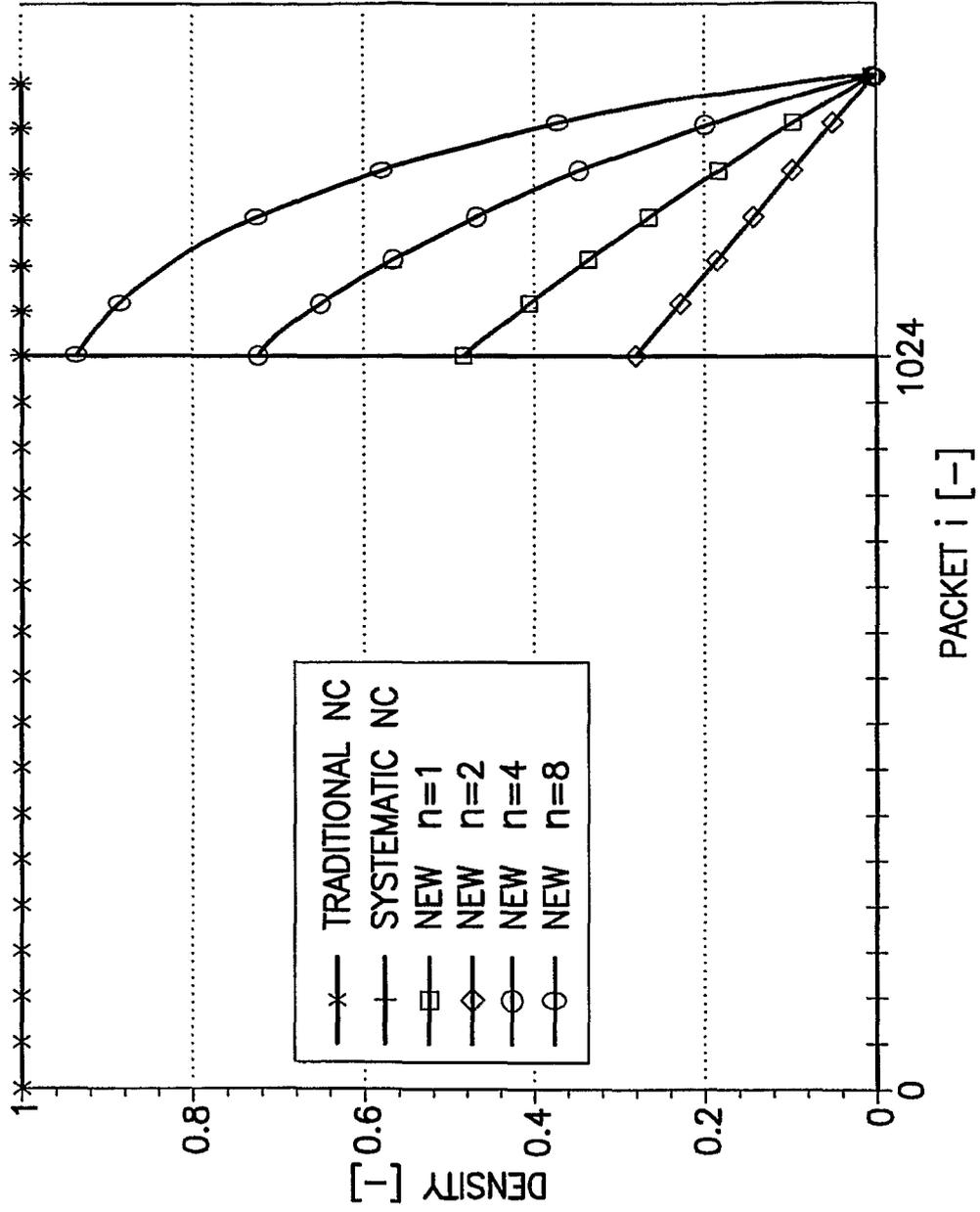


FIG.4

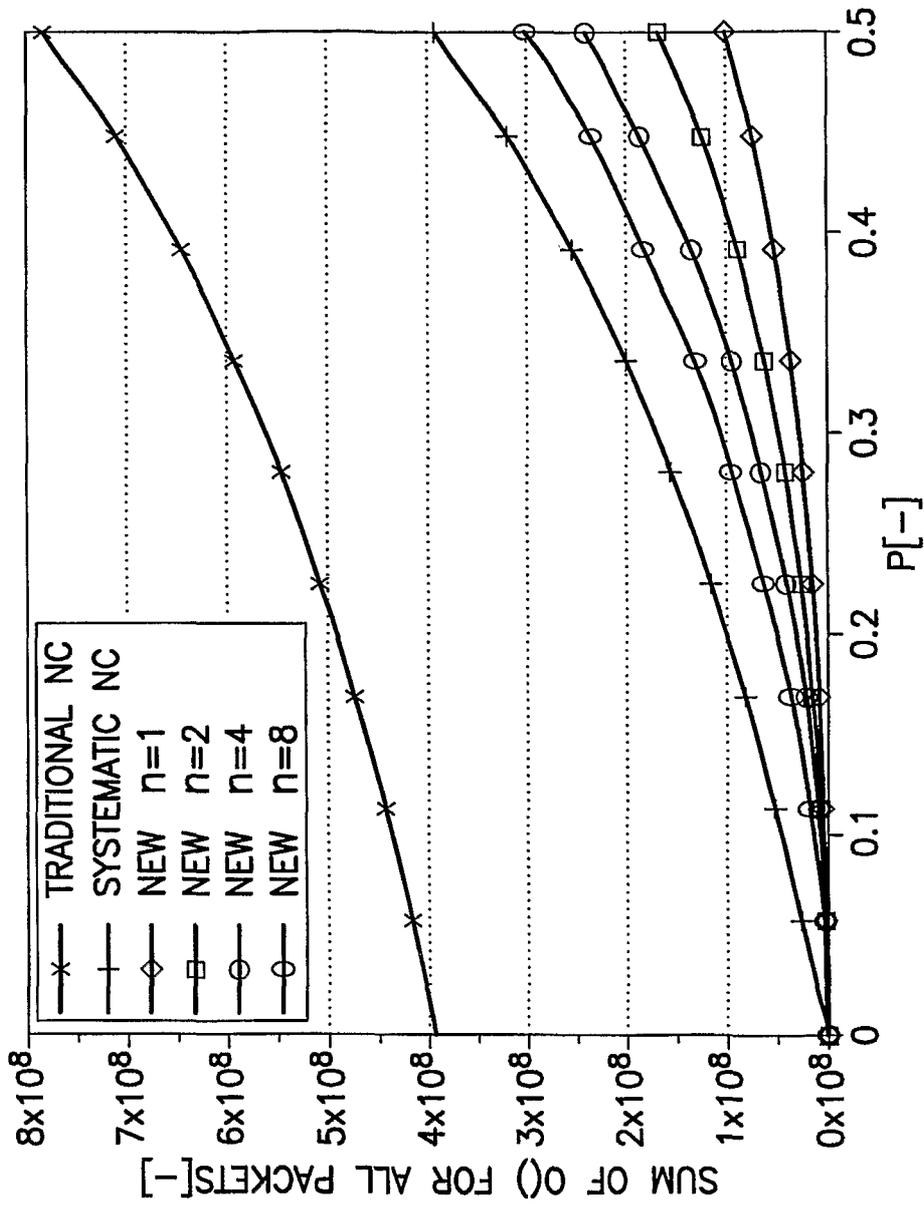


FIG.5

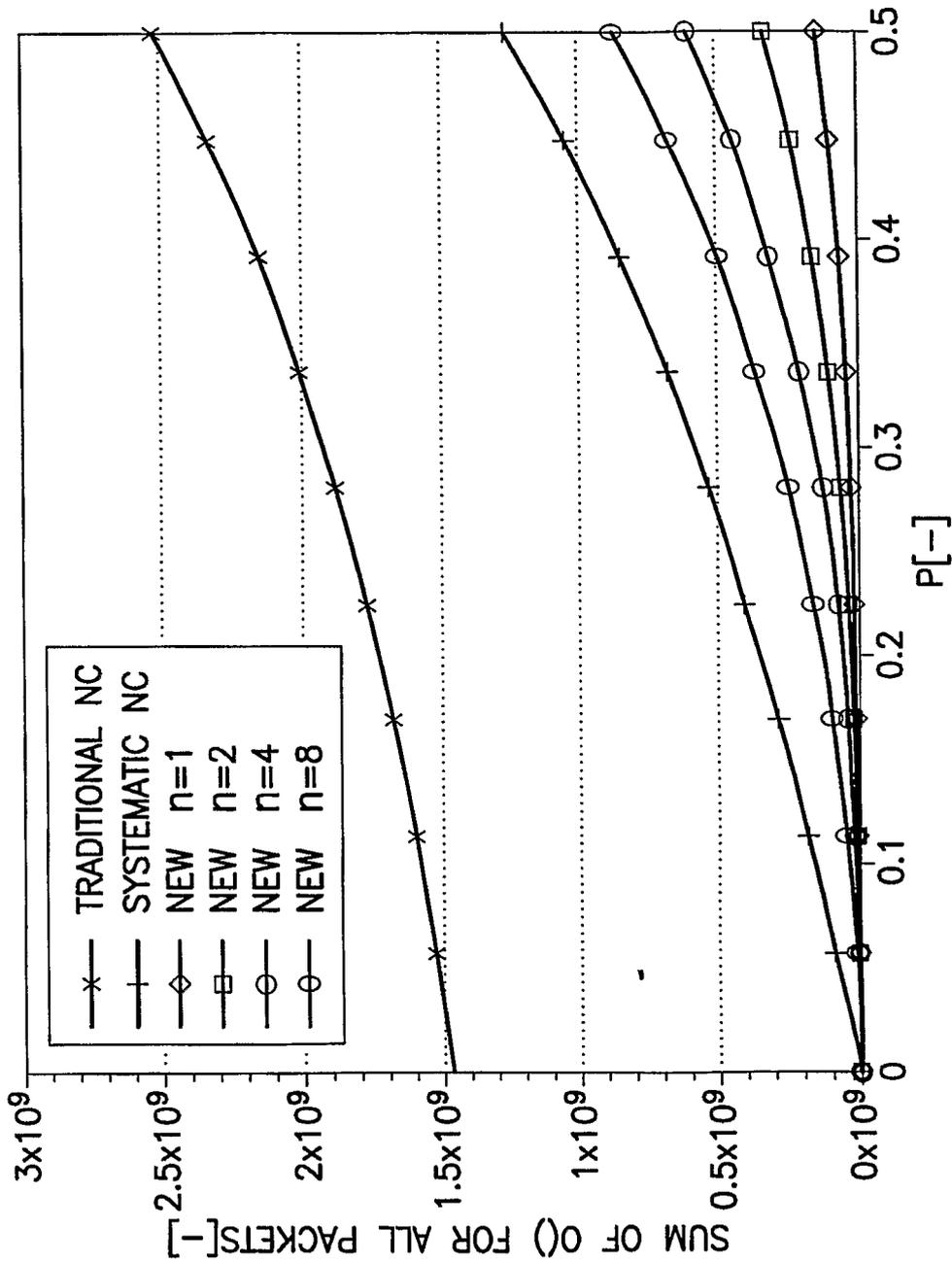


FIG.6

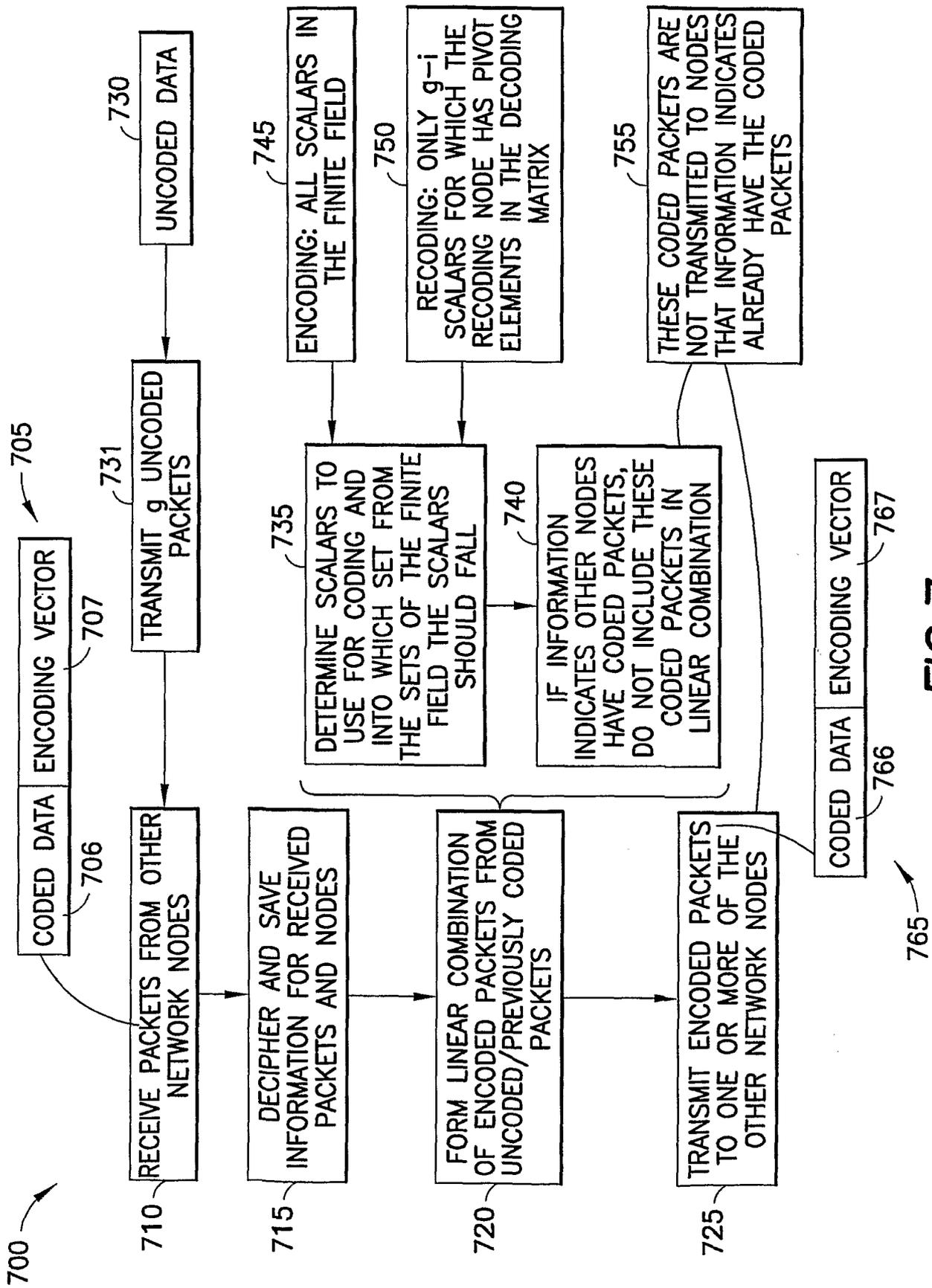


FIG. 7

800

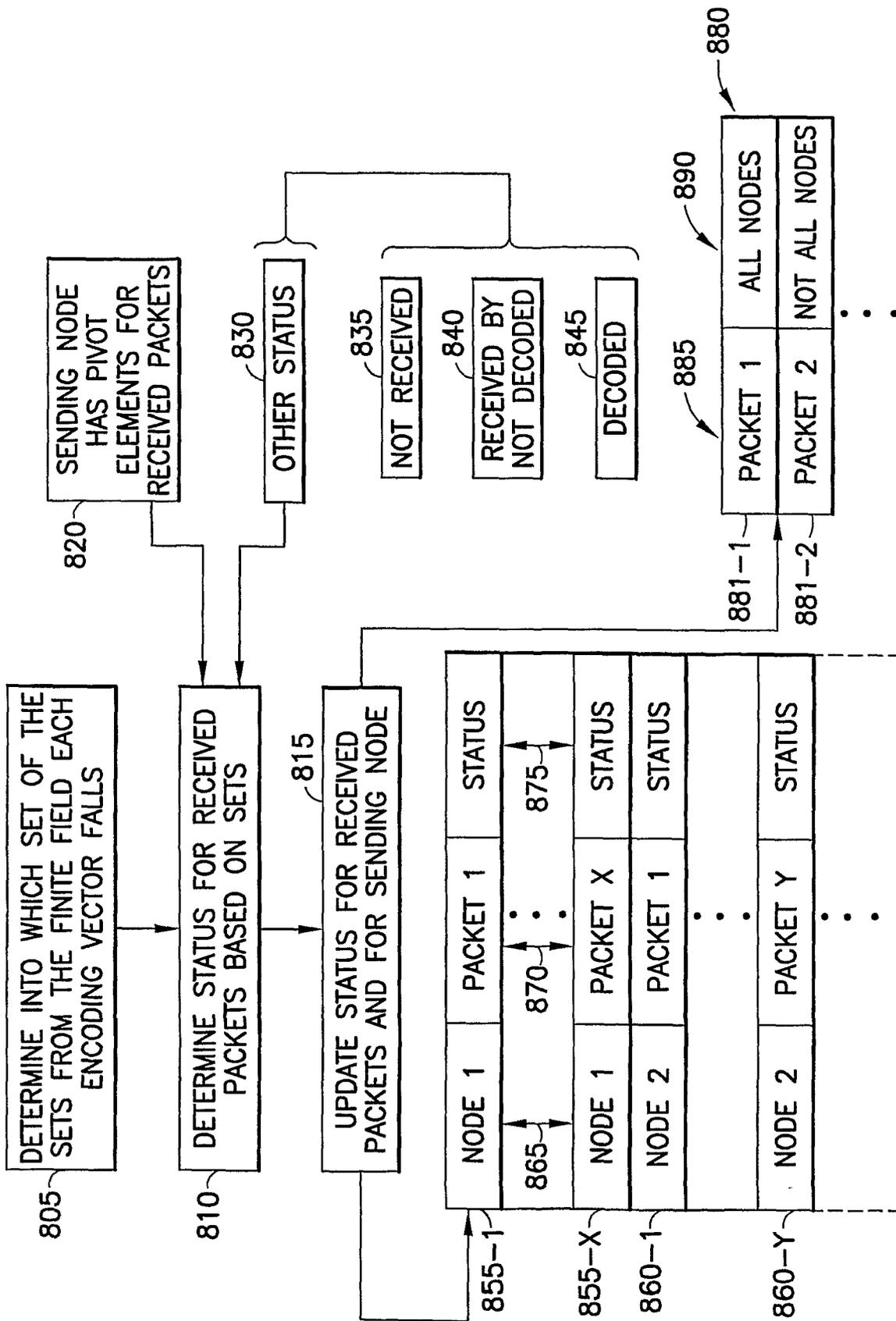


FIG.8

850

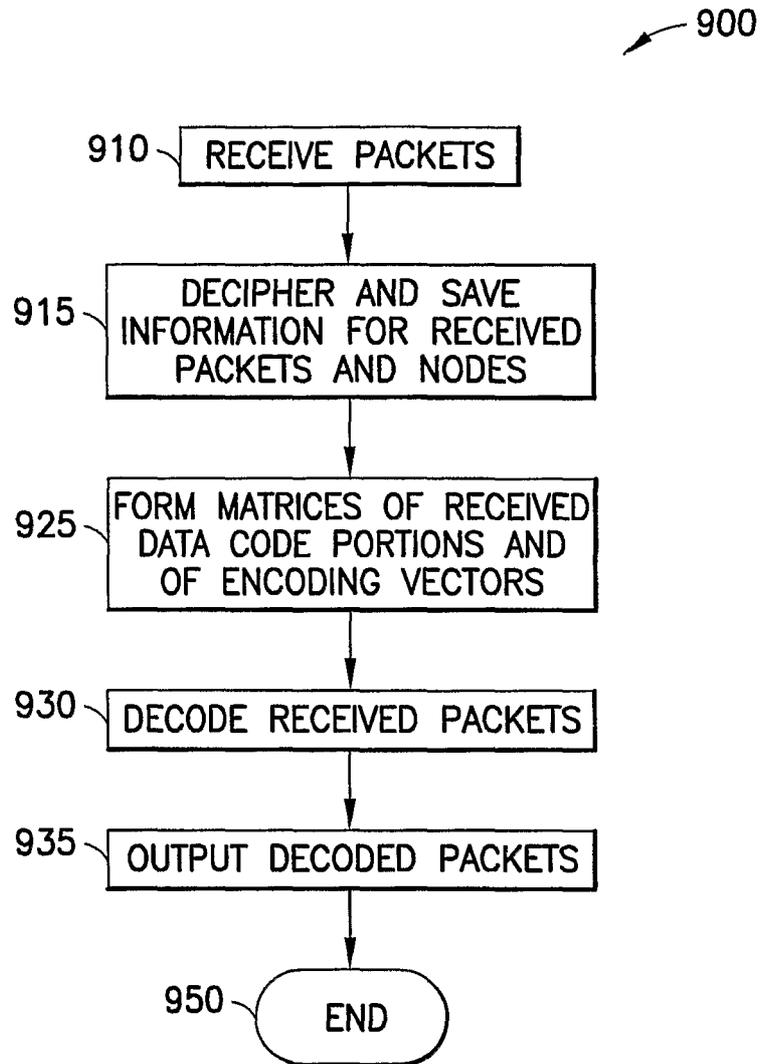


FIG.9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB201 0/051 208

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

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)

IPC:H04L

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

SE, DK, FI, NO classes as above

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

EPO-Intemal, PAJ, WPI data, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Shrader, B.; Jones, N.M.; , "Systematic wireless network coding," Military Communications Conference, 2009. MILCOM 2009. IEEE , vol., no., pp.1-7, 18-21 Oct. 2009; whole document; abstract; figure 1; Section II A Lines 25-35; Equations 2-3	1-24
Y	US 20100046371 A1 (SUNDARARAJAN JAY KUMAR ET AL), 25 February 2010 (2010-02-25); abstract; paragraphs [0043], [0048], [0200], [0214]; claim 6	1-24
A	Heide, J.; Pedersen, M.V.; Fitzek, F.H.P.; Larsen, T.; , "Network Coding for Mobile Devices - Systematic Binary Random Rateless Codes," Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on , vol., no., pp.1-6, 14-18 June 2009; whole document; abstract	1-24
A	WO 2008108708 A1 (ERICSSON TELEFON AB L M ET AL), 12 September 2008 (2008-09-12); whole document	1-24

 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

23 -11- 2010

Date of mailing of the international search report

24-1 1-2010Name and mailing address of the ISA/SE
Patent- och registreringsverket
Box 5055
S-102 42 STOCKHOLM

Facsimile No. +46 8 666 02 86

Authorized officer

Niclas Larsson

Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB201 0/051 208

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 20070081 23 A 1 (ERICSSON TELEFON AB L M ET AL), 18 January 2007 (2007-01 -18); whole document --	1-24
A	WO 20081 08700 A2 (ERICSSON TELEFON AB L M ET AL), 12 September 2008 (2008-09-1 2); whole document -- -----	1-24

Continuation of: second sheet

International Patent Classification (IPC)

H04L 72/56 (2006.01)

Download your patent documents at www.prv.se

The cited patent documents can be downloaded:

- From "Cited documents" found under our online services at www.prv.se
(English version)
- From "Anforda dokument" found under "e-tjanster" at www.prv.se
(Swedish version)

Use the application number as username. The password is **XAHOWDPIGK**.

Paper copies can be ordered at a cost of 50 SEK per copy from PRV InterPat (telephone number 08-782 28 85).

Cited literature, if any, will be enclosed in paper form.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/IB201 0/051 208

US	201 00046371	A 1	25/02/201 0	NONE			
WO	20081 08708	A 1	12/09/2008	EP	2 1 19086	A 1	18/1 1/2009
				JP	201 0520700	T	10/06/201 0
				US	20100088568	A 1	08/04/201 0
WO	20070081 23	A 1	18/01/2007	CA	261 3655	A 1	18/01/2007
				CN	10 12 13781	A	02/07/2008
				EP	19001 36	A 1	19/03/2008
				EP	19001 37	A 1	19/03/2008
				JP	2008545330	T	11/1 2/2008
				US	2008031 0409	A 1	18/1 2/2008
				US	2008021 251 0	A 1	04/09/2008
				US	771 0908	B2	04/05/201 0
				US	200901 47738	A 1	11/06/2009
				WO	20070081 63	A 1	18/01/2007
				WO	20070081 62	A 1	18/01/2007
WO	20081 08700	A 2	12/09/2008	CN	10 162281 0	A	06/01/201 0
				EP	2 1 15925	A2	11/1 1/2009
				JP	201 0521 838	T	24/06/201 0
				US	20100061 259	A 1	11/03/201 0