



(12) **Patentschrift**

(21) Aktenzeichen: **10 2012 203 653.9**
 (22) Anmeldetag: **08.03.2012**
 (43) Offenlegungstag: –
 (45) Veröffentlichungstag
 der Patenterteilung: **18.07.2013**

(51) Int Cl.: **H03M 13/00 (2012.01)**

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
**Deutsches Zentrum für Luft- und Raumfahrt e.V.,
 51147, Köln, DE**

(72) Erfinder:
**Garrammone, Giuliano, 80686, München, DE;
 Blasco, Francisco Lazaro, Dipl.-Ing., 80807,
 München, DE**

(74) Vertreter:
von Kreisler Selting Werner, 50667, Köln, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht
 gezogene Druckschriften:

US 2007 / 0 195 894 A1
WO 2006/ 038 054 A1

(54) Bezeichnung: **Verfahren zum Wiederherstellen von verloren gegangenen und/oder beschädigten Daten**

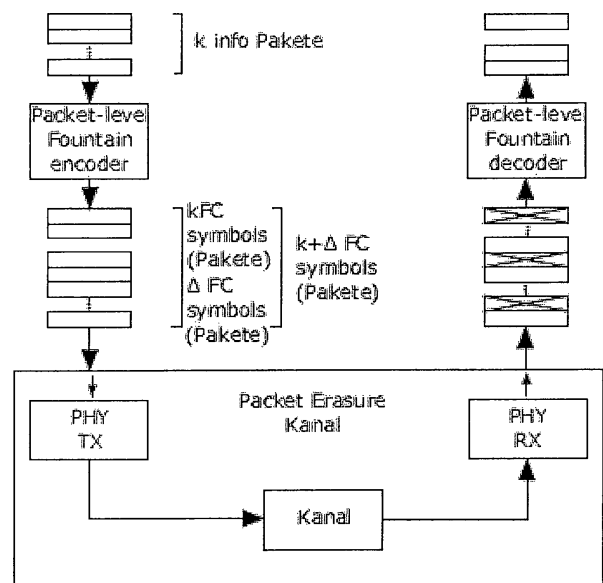
(57) Zusammenfassung: Verfahren zum Wiederherstellen von verloren gegangenen und/oder beschädigten Daten, die von einer Sendevorrichtung zu einer Empfängervorrichtung übertragen werden, wobei dieses Verfahren die Schritte aufweist:

Codieren der genannten Daten durch einen Encoder, der mit der Sendevorrichtung verbunden ist,

Übertragen der genannten Daten von der Sendevorrichtung zu der Empfängervorrichtung über einen Übertragungskanal und

Decodieren der genannten Daten durch eine Decoder, der mit der Empfängervorrichtung verbunden ist, wobei verloren gegangene und/oder beschädigte Daten während des Decodierens wiederhergestellt werden

wobei eine Erzeugermatrix zum Codieren und Decodieren verwendet wird, wobei Daten vor dem Decodieren in k Eingangssymbole der Länge L aufgeteilt werden, wobei die Ausgangssymbole des Encoders fragmentiert werden, um in die Maximum Transmission Unit (MTU) der Frames im Physical Layer zu passen, wobei während des Decodierens der Empfänger die empfangenen Fragmente der Ausgangssymbole behandelt, als wären sie Ausgangssymbole von b parallelen Encodern, deren Eingangssymbole k Datenpakete der Größe L/b Bits oder Bytes sind, wobei die empfangenen Fragmente decodiert werden unter Verwendung von b parallelen Decodern dadurch gekennzeichnet, dass die Operationen, die ausgeführt werden an den Gleichungen, die die zu lösenden b Gleichungssysteme gemeinsam haben, einmal anstelle von b mal ausgeführt werden, sodass die Decodierkomplexität verringert wird.



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum Wiederherstellen von verloren gegangenen und/oder beschädigten Daten.

[0002] Es ist bekannt, dass Daten während der Übertragung beispielsweise über einen rauschbehafteten Kanal durch verschiedene Fehlerkorrekturverfahren geschützt werden können. Zu diesem Zweck werden m Parity Pakete durch einen Encoder erzeugt, die k Informationspaketen hinzugefügt werden, so dass $n = k + m$ Codeword Pakete über den Kanal übertragen werden. Durch Verwenden der übertragenen Parity Informationen können verloren gegangene oder beschädigte Daten wiederhergestellt werden.

[0003] Ein Codierschema, das aus dem Stand der Technik bekannt ist, ist Fountain Coding. Fountain Coding kann beispielsweise auf packet Level angewandt werden und ist eine einfache und effiziente Technik, um in einem Kommunikationssystem eine verlässliche Übertragung sicher zu stellen. Die Grundprinzipien von Packet Level Fountain Coding sind in [Fig. 1](#) dargestellt.

[0004] Zuerst wird die zu übertragende Nachricht, z. B. eine Datei, aufgeteilt in k Infopakete von L Bits oder Bytes (Eingangssymbole für den Fountain Encoder) und codiert in $k + \Delta$ Fountain Code (FC) Symbole (Pakete von L Bits oder Bytes). Somit werden die $k + \Delta$ Pakete erzeugt durch den Paket Level Encoder und sind die Ausgangssymbole des Fountain Encoders. Δ ist der Overhead auf der Transmitterseite, d. h. die Anzahl an Paketen, zusätzlich zu den k die durch den Encoder erzeugt werden.

[0005] Zweitens, werden die $k + \Delta$ Fountain Encoder Symbole im Physical Layer (innerhalb des PHY Layer Frames) durch Fehlerkorrekturcodes (z. B. Turbo, LDPC...), Fehlererkennungscode (z. B. Cyclic Redundancy Check (CRC)) geschützt und sie werden übertragen.

[0006] Drittens wird auf jedes Paket auf der Empfängerseite Physical Layer Fehlerkorrektur angewandt und verbleibende Fehler werden durch einen Fehlererkennungscode detektiert. Wenn Fehler detektiert werden, wird das Paket als verloren angesehen und als Auslöschung gekennzeichnet. Somit sehen die Layer über dem Physical Layer das Kommunikationsmedium als einen Packet Erasure Channel (PEC), wo Pakete entweder korrekt empfangen werden oder verloren gehen.

[0007] Zuletzt stellt der Packet-Level Decoder die Original Nachricht wieder her, sofern eine ausreichende Menge an Paketen empfangen wurde.

[0008] Fountain Codes sind ratenlose Codes, d. h. $\Delta = 0, 1, 2, \dots$ was bedeutet, dass keine Grenze für die Anzahl von erzeugten Fountain Encoder Paketen existiert. Fountain Encoder Pakete werden erzeugt, bis der Packet-Level Decoder in der Lage ist, die ursprüngliche Nachricht wieder herzustellen. Insbesondere ist das Decodieren umso komplexer, je größer k ist. Daher ist die Auswahl der Paketgröße L vom Blickwinkel der Komplexität her wichtig.

[0009] Eine Klasse von einfachen und effektiven Fountain Codes ist beispielsweise die Linear Random Fountain Codes (LRFC) Klasse über Galois Felder höherer Ordnung [G. Liva, E. Paolini, und M. Chiani, "Performance versus Overhead for Fountain Codes over F_q ", IEEE Communications Letters, vol. 14, No. 2, Seiten 178–180, Februar 2010.]. Die Decodierkomplexität für LRFC ist kubisch über k , d. h. $O(k^3)$. Um kein komplexes Fountain Decoding zu haben, sollte der Faktor k , auf den die Daten aufgeteilt werden, nicht groß sein, sodass die Paketgröße L nicht klein sein sollte. Jedoch kann, damit jedes Fountain Coding Ausgangssymbol innerhalb eines PHY Layer Frames übertragen wird, die Paketgröße L die PHY Max Transmission Unit (MTU) nicht überschreiten und dies könnte zu einem großen k führen und somit zu einer hohen Decodingkomplexität.

[0010] Softwareimplementierungen von Packet-Level Encodern und Decodern sind besonders attraktiv, da sie keine hohen Anstrengungen für Design/Implementation erfordern und sie eine höhere Flexibilität gegenüber Hardware Implementierungen ermöglichen. Software Module können einfach in eine Hardware Architektur integriert werden, die nicht speziell designt wurde um Packet-Level Coding zu unterstützen.

[0011] Da Software Packet-Level Encoder und Decoder auch bestimmt sind, um auf energiebeschränkten Plattformen zu funktionieren (z. B. on-Board Einheiten von Raumschiffen, Landeschiffen, Rover, Orbiter in erdnahen Missionen und in Missionen im fernen Weltraum, Satelliten), sind effiziente Verfahren notwendig.

[0012] Wenn andere Codingschemata als Fountain Codes verwendet werden, kann dasselbe Problem auftreten, nämlich, dass die Decodingkomplexität sehr schnell ansteigt, wenn die Anzahl von k Informationspaketen erhöht wird (nämlich zu Ok^3). Ferner existiert nach wie vor das Problem, dass wenn die Informationspakete zu lang sind (was bedeutet, dass L zu groß ist) diese Pakete nicht in die Maximum Transmission Unit (MTU) im Physical Layer passen und daher fragmentiert werden müssen um in die MTU zu passen;

Ein Packet-Level Coding Schema mit Fountain Codes funktioniert wie folgt:

1. Daten z. B. eine DATEI, von DATEI_GRÖSSE Bytes wird aufgeteilt in k Pakete $P_i = 1, 2, \dots, k$ der Größe L Bits oder Bytes (Eingangssymbole für Fountain Code).
2. Der Packet-Level Fountain Encoder gibt $k + \Delta$ Pakete (C_j) der Größe L Bits oder Bytes (Ausgangssymbole für Fountain Code) gemäß dem folgenden System aus:

$$\begin{bmatrix} g_{11} & \cdots & g_{1k} \\ g_{21} & \cdots & g_{2k} \\ \vdots & \ddots & \vdots \\ g_{(k+\Delta)1} & \cdots & g_{(k+\Delta)k} \end{bmatrix} \begin{pmatrix} P_1 \\ \vdots \\ P_k \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_{k+\Delta} \end{pmatrix} \quad (1)$$

wobei die Koeffizienten der Matrix g_{ij} , mit $i = 1, 2, \dots, k + \Delta$ und $j = 1, 2, \dots, k$ zu einem Galois Feld (GF) der Ordnung $q \geq 2$ gehören. Die Anzahl der Ausgangssymbole die auf der rechten Seite gezeigt sind hängt von der Qualität des Übertragungskanal ab, wenn ein Fountain Code verwendet wird, da der Encoder fortwährend neue Ausgangssymbole ausgegeben wird, bis der Empfänger anzeigt, dass alle Daten wieder hergestellt wurden.

3. Auf der Empfängerseite müssen $m \geq k$ Fountain Code Symbole empfangen werden, um das Linearsystem (2) zu lösen, in dem k Pakete $P_i = 1, 2, \dots, k$ die Unbekannten sind. Es werden bezeichnet mit $J = \{j_1, j_2, \dots, j_m\}$ das Set von Indices der empfangenen Fountain Code Symbole. Die empfangenen Fountain Code Symbole sind somit bezeichnet durch $\{C_{j_1}, C_{j_2}, \dots, C_{j_m}\}$.

$$\begin{bmatrix} g_{j_1 1} & \cdots & g_{j_1 k} \\ g_{j_2 1} & \cdots & g_{j_2 k} \\ \vdots & \ddots & \vdots \\ g_{j_m 1} & \cdots & g_{j_m k} \end{bmatrix} \begin{pmatrix} P_1 \\ \vdots \\ P_k \end{pmatrix} = \begin{pmatrix} C_{j_1} \\ C_{j_2} \\ \vdots \\ C_{j_m} \end{pmatrix} \quad (2)$$

[0013] Das Linearsystem (2) kann gelöst werden, durch Gaußsche Elimination mit einer kubischen Komplexität in k .

[0014] Die minimale Dimension des Systems (2) ist $k \times k$, um eine Wahrscheinlichkeit ungleich null für ein erfolgreiches Decodieren zu haben. Bei gegebener DATEI-GRÖSSE bestimmt der Wert der Paketgröße L den Wert von k , der $k = \text{ceil}(\text{DATEI_GRÖSSE}/L)$ ist. Somit sind k und die Decodingkomplexität umso kleiner, je größer L ist. Jedoch kann, um jedes Fountain Code Symbol innerhalb eines PHY Layer Frames zu übertragen, die Paketgröße L nicht die PHY MTU übersteigen, und dies könnte zu einem großen k führen und somit zu einer hohen Decodingkomplexität (großes System zu lösen mit Gaußscher Elimination).

[0015] Eine Lösung, um der Anforderung im PHY Layer durch das MTU und dem Wunsch ein großes L im packet-Layer zu haben gerecht zu werden, die zu einer geringeren Fountain Decodingkomplexität führt, ist Fragmentation: Ein erstes Schema, in dem Fragmentation verwendet wird, ist [Fig. 2](#) dargestellt.

[0016] Fragmentation bedeutet, dass jedes Fountain Coding Symbol C_j , $j = 1, 2, \dots, k + \Delta$ gesendet wird innerhalb b PHY Layer Frames mit $b \geq 2$.

[0017] Somit wird jedes Fountain Code Symbol (Paket der Größe L Bits oder Bytes) aufgeteilt in b Fragmente C_j^z , $z = 1, \dots, b$ (Pakete der Größe L/b Bits oder Bytes) und jedes von ihnen wird mit einem PHY Layer Frame übertragen.

[0018] Auf der Empfängerseite findet zuerst die Fragmentierung statt durch Sammeln der b PHY Frames für jedes Fountain Encoding Symbol C_j , dann wird Fountain Decoding durch Lösen des Systems (2) durchgeführt. Dieses Vorgehen erlaubt es große L zu haben und somit ein kleines k , was zu brauchbarer Decodingkomplexität führt. Der Nachteil dieses Vorgehens ist, dass ein Fountain Coding Symbol verloren geht, wenn wenigstens ein Frame (Fragment) verloren geht.

[0019] Die Alternative der Fragmentation (was in den Schritten 1–3 angenommen wird) ist die DATEI aufzuteilen in Pakete der Größe L/b Bits oder Bytes und jedes von ihnen innerhalb eines PHY Layer Frames zu senden. Schema 2 in [Fig. 3](#) zeigt dieses Vorgehen, das auf die gleiche DATEI_GRÖSSE von $(k \cdot L)$ Bytes, die in [Fig. 2](#) dargestellt ist, angewandt wird (um einen fairen Vergleich zu haben). Somit ist keine Fragmentierung notwendig, aber ein größeres System von mindestens $(b \cdot k \times b \cdot k)$ muss auf der Empfängerseite gelöst werden (höhere Komplexität). Es ist zu beachten, dass um die gleiche DATEI Größe von $k \cdot L$ Bytes und Pakete von L/b Bytes beizubehalten, wir $b \cdot k$ Pakete benötigen. Mit anderen Worten ist die Symbolgröße ohne Fragmentierung L/b , während $b \cdot k$ Symbole benötigt werden, um die gleiche DATEI Größe zu erreichen.

[0020] Zusammenfassend sind die Fountain Coding Schemata um z. B. eine DATEI von DATEI_GRÖSSE Bits oder Bytes zu liefern, gemäß dem Stand der Technik:

1. Fragmentation mit jedem FC Symbol gesendet innerhalb b PHY Layer Frames: Schema 1 in [Fig. 2](#).
2. Keine Fragmentierung mit jedem FC Symbol übertragen innerhalb eines PHY Layer Frames ($L \leq MTU$) : Schema 2 in [Fig. 3](#)

[0021] Zusätzlich zu dem oben genannten Stand der Technik können Informationen über packet level Fountain Encoding gefunden werden in [J. Byers, M. Luby, M. Mitzenmacher, und A. Rege, "A digital fountain approach to reliable distribution of bulk data," SIGCOMM Comput. Commun. Rev., vol. 28, no. 4, Seiten 56–67, Okt. 1998.].

[0022] Im Folgenden werden einige mathematische Erklärungen bezüglich der Nachteile der oben genannten Methoden aus dem Stand der Technik dargestellt:

Wenn die Menge an zu übertragenden Daten groß ist, ist das Schema in [Fig. 3](#) (Schema 2) ohne PHY Layer Fragmentierung nicht durchführbar/praktikabel aufgrund hoher Decodierkomplexität und somit Decodierzeit. Da $k \sim \text{DATEI_GRÖSSE}/L$ und $L < MTU$, kann k sehr groß sein und somit auch die Decodierkomplexität.

[0023] Der Nachteil des Schemas mit PHY Layer Fragmentierung (Schema 1) ist ein großer Performance Verlust im Vergleich zu dem Schema ohne PHY Fragmentation (Schema 2). Für diesen Fall, $L > MTU$ ist somit k kleiner als in Schema 2, aber b Frames werden für jedes FC Symbol gesendet und ein FC Symbol geht verloren wenn mindestens einer von b Frames verloren geht.

[0024] Eine Obergrenze für die Versagenswahrscheinlichkeit für ein System ohne Fragmentation (Schema 2 in [Fig. 3](#)) ist hiernach angegeben:

$$p_f^{NOfrag} = \sum_{i=0}^{bk-1} \binom{bk+\Delta}{i} (1-\varepsilon)^i \varepsilon^{bk+\Delta-i} + \sum_{i=bk}^{bk+\Delta} \binom{bk+\Delta}{i} (1-\varepsilon)^i \varepsilon^{bk+\Delta-i} p_e(i-bk, q)$$

[0025] Eine Obergrenze der Versagenswahrscheinlichkeit für ein System mit Fragmentierung (Schema 1 in [Fig. 2](#)) ist hiernach angegeben:

$$p_f^{frag} = \sum_{i=0}^{k-1} \binom{k+\Delta}{i} (1-\varepsilon')^i (\varepsilon')^{k+\Delta-i} + \sum_{i=k}^{k+\Delta} \binom{k+\Delta}{i} (1-\varepsilon')^i (\varepsilon')^{k+\Delta-i} p_e(i-k, q)$$

wobei ε die Frameauslöschungswahrscheinlichkeit ist (die identisch mit der Paketauslöschungswahrscheinlichkeit für Schema 2 ist) und ε' die FC Symbol Auslöschungswahrscheinlichkeit für Schema 1 ist. Insbesondere ist

$$\varepsilon' = 1 - (1 - \varepsilon)^b$$

in Schema 2. Für den Fall eines Versagens des Systems empfängt der User weniger als b_k (k in Schema 1) FC Symbole oder es erhält wenigstens b_k (k in Schema 1) FC Symbole und das Decodieren versagt mit einer Wahrscheinlichkeit die berechnet wurde in "G. Liva, E. Paolini, und M. Chiani, "Performance versus Overhead for Fountain Codes over F_q ", IEEE Communications Letters, vol. 14, No. 2, Seiten 178–180, Februar 2010". Die Wahrscheinlichkeit eines Decodierversagens ist die Wahrscheinlichkeit, dass eine Matrix der Größe $(k + \delta) \times k$ mit Zufallselementen in $GF(q)$ keinen vollen Rang hat. Die Obergrenze dieser Wahrscheinlichkeit ist:

$$p_\varepsilon(\delta, q) = \frac{1}{q-1} q^{-\delta}$$

wobei q die Ordnung des GF ist, auf dem der LRFC aufgebaut ist und δ der Overhead auf der Empfängerseite ist, d. h. die Anzahl der gesammelten Gleichungen zusätzlich zu k . Die Komplexität des Schemas 1 ist $O(k^3)$, daher um Faktor b^3 besser als Schema 2 das eine Komplexität von $O(b^3k^3)$ hat.

[0026] Fig. 4 zeigt den Performanceverlust aufgrund von Fragmentierung für ein realistisches Beispiel, das später dargestellt wird. Die Wahrscheinlichkeit für ein Versagen für Schema 1 und Schema 2 gegenüber dem Overhead Δ_F im Hinblick auf die Anzahl der zusätzlichen Frames, die durch den Sender auf den Kanal übertragen werden, ist dargestellt. Es ist sichtbar wie, wenn Schema 1 angewandt wird, der Sender viel mehr Overheadsymbole Δ_F im Vergleich zu Schema 2 übertragen muss.

[0027] Gemäß der Methode, die in DE 10 2011 115 100.5, die noch nicht veröffentlicht wurde, dargestellt ist, werden Daten übertragen von einer Sendevorrichtung zu einer Empfängervorrichtung wobei Daten durch einen Encoder, der mit der Sendevorrichtung verbunden ist, codiert werden.

[0028] Gemäß diesem Verfahren werden Daten vor dem Codieren in k Eingangssymbole aufgeteilt, von denen jedes die Länge L hat. Diese Eingangssymbole werden dem Encoder zugeführt, wobei die Ausgangssymbole des Encoders fragmentiert werden, um in die Maximum Transmission Unit (MTU) der Frames im Physical Layer zu passen.

[0029] Diese Methode ist dadurch gekennzeichnet, dass während des Decodierens der Empfänger die empfangenen Fragmente der Ausgangssymbole behandelt als wären sie die Ausgangssymbole von b parallelen Encodern, deren Eingangssymbole k Datenpakete der Größe L/b Bits oder Bytes sind.

[0030] Mit anderen Worten wird das Decodieren wie oben beschrieben durchgeführt, obwohl tatsächlich die Daten auf der Senderseite vor dem Codieren in k Eingangssymbole der Größe L aufgeteilt wurden und nicht in b Sets von k Eingangssymbolen der Größe L/b Bits oder Bytes.

[0031] Dieses Verfahren nutzt die Tatsache, dass Fragmente desselben Typs gemäß linearer Gleichungen, die in der Erzeugermatrix dargestellt sind, zueinander in Beziehung stehen. Mit anderen Worten verwendet das Verfahren Informationen, die bereits in den Fragmenten vorhanden sind, die über den Übertragungskanal übertragen werden. Es ist nicht notwendig, das Codierschema zu ändern. Gemäß diesem Verfahren muss nur das Decodierschema wie oben beschrieben angepasst werden.

[0032] Das beschriebene Verfahren beinhaltet Fragmentierung und reduziert die Komplexität des oben genannten Schemas 2 aus dem Stand der Technik. Ferner wird eine Verringerung des Performanceverlusts für Codingschemata erreicht, wenn eine Fragmentierung notwendig ist. Daher wird der Performanceverlust des Schemas 1 gemäß dem Stand der Technik reduziert. Daher funktioniert das erfindungsgemäße Verfahren besser als beide Schemata, wenn Physical Layer Fragmentierung notwendig ist.

[0033] In diesem Verfahren behandelt der Empfänger für $z = 1, \dots, b$ die empfangenen Fragmente, als wären sie erzeugt worden gemäß

$$\begin{bmatrix} g_{11} & \cdots & g_{1k} \\ g_{21} & \cdots & g_{2k} \\ \vdots & \ddots & \vdots \\ g_{(k+\Delta)1} & \cdots & g_{(k+\Delta)k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_1^z \\ C_2^z \\ \vdots \\ C_{k+\Delta}^z \end{pmatrix} \quad (3)$$

[0034] Dabei ist der linke Part der Gleichung die Erzeugermatrix, P_1^z, \dots, P_k^z die Eingangssymbole der Länge L/b Bits oder Bytes und $C_1^z, C_2^z, \dots, C_{k+\Delta}^z$ sind die Ausgangssymbole der Länge L/b Bits oder Bytes, und die empfangenen Fragmente werden decodiert durch Verwendung b paralleler Decoder gemäß:

$$\begin{bmatrix} g_{j_1^z 1} & \cdots & g_{j_1^z k} \\ g_{j_2^z 1} & \cdots & g_{j_2^z k} \\ \vdots & \ddots & \vdots \\ g_{j_m^z 1} & \cdots & g_{j_m^z k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_{j_1^z}^z \\ C_{j_2^z}^z \\ \vdots \\ C_{j_m^z}^z \end{pmatrix} \quad (4)$$

wobei $J^z = \{j_1^z, j_2^z, \dots, j_m^z\}$ das Set von Indices der empfangenen Fragmente für den z -ten Decoder bezeichnen, sodass die empfangenen Fragmente bezeichnet werden durch $\{C_{j_1^z}^z, C_{j_2^z}^z, \dots, C_{j_m^z}^z\}$ für den z -ten Decoder.

[0035] Daher wird nicht das gesamte Ausgangssymbol C_j verworfen, wenn wenigstens ein Frame (Fragment) verloren gegangen ist, sondern die Tatsache wird genutzt, dass Fragmente desselben Typs z zueinander durch lineare Gleichungen in Beziehung stehen.

[0036] Eine Obergrenze für die Versagenswahrscheinlichkeit des vorgeschlagenen Schemas wurde wie folgt abgeleitet. Zunächst ist die Versagenswahrscheinlichkeit für jeden Fountain Decoder (ausgehend von unabhängigen Frameauslöschungen) für $z = 1, \dots, b$

$$p_f^z = \sum_{i=0}^{k-1} \binom{k+\Delta}{i} (1-\varepsilon)^i (\varepsilon)^{k+\Delta-i} + \sum_{i=k}^{k+\Delta} \binom{k+\Delta}{i} (1-\varepsilon)^i (\varepsilon)^{k+\Delta-i} p_e(i-k, q)$$

wobei ε die Frame-Auslöschungswahrscheinlichkeit ist. Damit jeder Fountain Decoder versagt, empfängt der User weniger als k seiner FC-Ausgangssymbole oder es empfängt wenigstens k von seinem FC-Ausgangssymbolen und das Decodieren versagt mit einer Wahrscheinlichkeit, deren obere Grenze gegeben ist durch p_e

$$p_e(\delta, q) = \frac{1}{q-1} q^{-\delta}$$

[0037] Daher ist eine Obergrenze für die Versagenswahrscheinlichkeit des Systems

$$p_f = 1 - (1 - p_f^z)^b$$

[0038] Fig. 5 zeigt die Versagenswahrscheinlichkeit für die vorgestellten Fountain Coding Schemata gegenüber dem Overhead Δ_F im Hinblick auf die Anzahl der zusätzlichen Frames, die durch den Sender auf dem Kanal übertragen werden für das Beispiel, das in der o. g. Patentanmeldung dargestellt ist.

[0039] Es ist zu beachten, dass der Fragmentierungsverlust mit Schema 1 (Stand der Technik) ungefähr 284 Frames beträgt, während die vorgeschlagene Lösung aus DE 10 2011 115 100.5 einen Fragmentierungsverlust von nur ~40 Frames hat, während die Komplexität um einen Faktor $b^2 = 16$ im Vergleich zu dem Schema ohne Fragmentierung (Schema 2) reduziert wird.

[0040] Im Folgenden wird die Decodierkomplexität des vorgeschlagenen Schemas untersucht. Insbesondere ist die Decodierkomplexität, die Komplexität für das Lösen von b Gleichungssystemen mit Gauss'scher Elimination (GE), somit ein System für jeden Fragmenttyp z , $z = 1, \dots, b$. Die Komplexität ist angegeben durch die Anzahl elementarer Additionen und Multiplikationen in dem Galois Feld, über dem der Code generiert wurde, die notwendig sind, um die Systeme zu lösen. Für jedes der b Systeme ist die folgende Anzahl an Additionen bezeichnet durch N_a^{GE} und Multiplikationen bezeichnet durch N_m^{GE} notwendig, um ein System der Größe $(k + \delta)k$ zu lösen:

$$N_a^{GE}(k, \delta, L, b, q) = \frac{k(k-1)[3(k+\delta)-k-1]}{6} + \left[\frac{k[2(k+\delta)-k-1]}{2} + \frac{k(k-1)}{2} \right] \frac{\frac{L}{b}}{\log_2 q}$$

$$N_m^{GE}(k, \delta, L, b, q) = \frac{k(k+1)[3(k+\delta)-k-2]}{6} + \left[\frac{k[2(k+\delta)-k-1]}{2} + \frac{k(k+1)}{2} \right] \frac{\frac{L}{b}}{\log_2 q},$$

wobei δ die Anzahl an Gleichungen zusätzlich zu k ist, was den Overhead auf der RX-Seite darstellt.

[0041] Der Einfachheit halber wird angenommen, dass jedes der b Systeme auf der RX-Seite dasselbe δ hat. k ist für jedes System identisch.

[0042] $\log_2 q$ ist die Anzahl der Elemente des Galois Feld der Ordnung q in jedem Paket der Größe L/b Bits (FC "virtuelles" Ausgangssymbol).

[0043] Daher ist die Anzahl der Operationen, die in der vorgeschlagenen Lösung notwendig ist und bezeichnet wird als $N_a^{\text{TOT-state-of-the-art}}$ und $N_m^{\text{TOT-state-of-the-art}}$:

$$N_a^{\text{TOT-state-of-the-art}} = bN_a^{GE}$$

$$N_m^{\text{TOT-state-of-the-art}} = bN_m^{GE}$$

[0044] Es ist zu beachten, dass die exakte Komplexität des vorgeschlagenen Algorithmus dargestellt wurde. Aus den obigen Gleichungen ist leicht zu erkennen, dass die asymptotische Komplexität der vorgeschlagenen Lösung kubisch ist in $k \sim bO(k^3) = O(b(k^3))$.

[0045] US 2007/0195894 A1 beschreibt ein Verfahren zum Wiederherstellen verlorengegangener Daten, die über einen Übertragungskanal übermittelt werden. Zum Codieren und Decodieren wird eine Erzeugermatrix verwendet, wobei Daten vor dem Decodieren in k Eingangssymbole der Länge L aufgeteilt werden.

[0046] Ähnliche Merkmale sind in der Druckschrift WO 2006/038054 A1 beschrieben.

[0047] Es ist eine Aufgabe der vorliegenden Erfindung, ein Verfahren zum Wiederherstellen verloren gegangenen und/oder beschädigten Daten bereitzustellen, dass eine verbesserte Decodierperformance und/oder eine reduzierte Decodierkomplexität aufweist.

[0048] Diese Aufgabe wird gelöst durch die Merkmale des Anspruchs 1 der Erfindung.

[0049] Gemäß der erfinderischen Methode werden Daten übertragen von einer Sendevorrichtung zu einer Empfängervorrichtung wobei Daten durch einen Encoder, der mit der Sendevorrichtung verbunden ist, codiert werden. Daten werden über einen Übertragungskanal übertragen, der beispielsweise ein Broadcasting-Netzwerk in einem Satellitenszenario sein kann. Jeder andere geeignete Übertragungskanal kann verwendet werden. Die übertragenen Daten werden decodiert durch einen Decoder, der mit der Empfängervorrichtung verbunden ist, wobei verloren gegangene oder beschädigte Daten während des Decodierens wiederhergestellt werden. Codieren und Decodieren wird durchgeführt unter Verwendung eines jeglichen geeigneten Codes, der z. B. ein linearer Blockcode, ein Fountaincode wie LT, raptor, LRFC etc. sein kann. All diese Codes haben gemeinsam, dass für das Codieren und Decodieren eine parity check Matrix verwendet wird.

[0050] Gemäß dem erfindungsgemäßen Verfahren werden Daten vor dem Decodieren in k Eingangssymbole aufgeteilt, von denen jedes die Länge L hat. Diese Eingangssymbole werden dem Encoder zugeführt, wobei die Ausgangssymbole des Encoders fragmentiert werden, um in die Maximum Transmission Unit (MTU) der Frames im Physical Layer zu passen.

[0051] In dem erfindungsgemäßen Verfahren behandelt der Empfänger empfangene Fragmente der Ausgangssymbole, als wenn sie Ausgangssymbole von b parallelen Encodern wären, deren Eingangssymbole k Datenpakete der Größe L/b Bits oder Bytes sind.

[0052] Mit anderen Worten wird das Decodieren, wie oben beschrieben, durchgeführt, obwohl tatsächlich die Daten auf der Senderseite vor dem Codieren in k Eingangssymbole der Größe L aufgeteilt wurden und nicht in b Sets von k Eingangssymbolen der Größe L/b Bits oder Bytes.

[0053] Daher nutzt das erfindungsgemäße Verfahren die Tatsache das Fragmente desselben Typs gemäß linearer Gleichungen, die in der Erzeugermatrix dargestellt sind, zueinander in Beziehung stehen. Mit anderen Worten verwendet die Erfindung Informationen, die bereits in den Fragmenten vorhanden sind, die über den Übertragungskanal übertragen werden. Es ist nicht notwendig, das Codierschema zu ändern. Es ist bevorzugt, dass gemäß der Erfindung nur das Decodierschema wie oben beschrieben angepasst werden muss.

[0054] Gemäß dem erfindungsgemäßen Verfahren werden die empfangenen Fragmente decodiert unter Verwendung von b parallelen Decodern.

[0055] Das erfindungsgemäße Verfahren ist dadurch gekennzeichnet, dass Operationen, die auf die Gleichungen angewendet werden, die die b Gleichungssysteme, die zu lösen sind, gemeinsam haben, nur einmal anstatt b mal ausgeführt werden, sodass die Decodierkomplexität verringert wird.

[0056] Somit stellt das erfindungsgemäße Verfahren ein effizientes Fountain Decoding Schema dar, durch das die Decodierkomplexität verringert werden kann, wie auch später im Detail gezeigt wird. Die Decodierkomplexität verglichen mit dem Decodieralgorithmus aus dem Stand der Technik kann bis zu einem Faktor b reduziert werden. Wie weiter unten dargestellt wird, ist die Decodierkomplexität des Decodieralgorithmus aus dem Stand der Technik, die Komplexität Fountain Decoding b mal auf Matrizen mit einer Mindestgröße von $(k + \delta) \times k$ auszuführen, während die Komplexität des erfindungsgemäßen Decodierverfahrens, die Komplexität ist, Fountain Decoding b mal bei kleineren Matrizen der Größe $(k - r + \delta) \times (k - r)$ auszuführen plus lediglich einen zusätzlich Fountain Decoding Schritt bei einer Matrix der Größe $r \times r$. Die Verringerung der Decodierkomplexität wird weiter unten im Detail erläutert.

[0057] Der vorgeschlagene Decodieralgorithmus kann mit jedem Fountain Code, wie LT, Raptor, LRFC, etc. verwendet werden.

[0058] Es ist bevorzugt, dass der Empfänger für $z = 1, \dots, b$ die empfangenen Fragmente behandelt, als wären sie erzeugt worden gemäß

$$\begin{bmatrix} g_{11} & \cdots & g_{1k} \\ g_{21} & \cdots & g_{2k} \\ \vdots & \ddots & \vdots \\ g_{(k+\Delta)1} & \cdots & g_{(k+\Delta)k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_1^z \\ C_2^z \\ \vdots \\ C_{k+\Delta}^z \end{pmatrix} \quad (3)$$

[0059] Dabei ist der linke Part der Gleichung die Erzeugermatrix, P_1^z, \dots, P_k^z die Eingangssymbole der Länge L/b Bits oder Bytes und $C_1^z, C_2^z, \dots, C_{k+\Delta}^z$ sind die Ausgangssymbole der Länge L/b Bits oder Bytes, und die empfangenen Fragmente werden decodiert durch Verwendung b paralleler Decoder gemäß:

$$\begin{bmatrix} g_{j_1^z 1} & \cdots & g_{j_1^z k} \\ g_{j_2^z 1} & \cdots & g_{j_2^z k} \\ \vdots & \ddots & \vdots \\ g_{j_m^z 1} & \cdots & g_{j_m^z k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_{j_1^z}^z \\ C_{j_2^z}^z \\ \vdots \\ C_{j_m^z}^z \end{pmatrix} \quad (4)$$

wobei $J^z = \{j_1^z, j_2^z, \dots, j_m^z\}$ das Set von Indices der empfangenen Fragmente für den z -ten Decoder bezeichnen, sodass die empfangenen Fragmente bezeichnet werden durch $\{C_{j_1^z}^z, C_{j_2^z}^z, \dots, C_{j_m^z}^z\}$ für den z -ten Decoder.

[0060] Das erfindungsgemäße Verfahren kann auch in Kombination mit anderen Codes als Fountain Codes, z. B. Linear Block Codes verwendet werden.

[0061] Wenn beispielsweise ein Fountain Coding Schema verwendet wird, könnte das erfindungsgemäße Verfahren die folgenden Schritte aufweisen:

1. Ein Block teilt Daten (z. B. eine DATEI) von DATEI_GRÖÖSE Bytes in k Pakete $P_i, i = 1, 2, \dots, k$ der Größe L Bytes (Eingangssymbole für den Fountain Code).
2. Ein Packet-Level Fountain Encoder, der $k + \Delta$ Pakete C_j der Größe L Bytes gemäß dem System in (1) ausgibt.
3. Ein Block teilt jede Linearkombination C_j der Größe L Bytes in b Fragmente $C_j^z, z = 1, \dots, b$ der Größe L/b Bytes die, in die MTU passen, auf (siehe [Fig. 2](#)).

[0062] Der Empfänger sammelt Frames, die die Fragmente $C_j^z, j = 1, 2, \dots, m$ und beinhalten. Die Idee ist, zu denken, dass diese Fragmente, die FC Ausgangssymbole von b parallelen Fountain Encodern sind, deren Eingangssymbole k Pakete der Größe L/b Bytes sind.

[0063] Der Empfänger denkt, dass die k Eingangssymbole der Größe L Bits oder Bytes in Schritt 1 aufgeteilt wurden in b Sets von k Eingangssymbolen der Größe L/b Bytes $\{P_i^z\}_{i=1}^k$.

[0064] Fountain Encoding wird dann auf jedes Set angewandt, sodass die b Sets von Ausgangssymbolen $\{C_j^z\}_{j=1}^{k+\Delta}$ erzeugt werden.

[0065] Daher denkt für $z = 1, \dots, b$ der Empfänger, dass das Codieren durchgeführt wurde gemäß der folgenden Gleichung,

$$\begin{bmatrix} g_{11} & \cdots & g_{1k} \\ g_{21} & \cdots & g_{2k} \\ \vdots & \ddots & \vdots \\ g_{(k+\Delta)1} & \cdots & g_{(k+\Delta)k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_1^z \\ C_2^z \\ \vdots \\ C_{k+\Delta}^z \end{pmatrix} \quad (3)$$

[0066] Es ist zu beachten, dass die Koeffizienten der b virtuellen Encoder die gleichen sind, sodass der Encoder in der Praxis nicht modifiziert wurde. Diese Art von dem Encoding Prozess zu denken führt zu dem folgenden Decodierschema:

4. Ein Empfänger sammelt die Ausgangssymbole von jedem Ausgangsset (virtuell generiert durch die b Fountain Encoder) und führt b Fountain Decoding parallel aus, wenn $m \geq k$ Ausgangssymbole aus jedem Ausgangsset empfangen werden. Jeder Fountain Decoder muss das folgende Linearsystem lösen, wo bezeichnet wird mit $J^z = \{j_1^z, j_2^z, \dots, j_m^z\}$ das Set von Indices der empfangenen FC Symbole für den z -ten Fountain Decoder. Die empfangenen FC Symbole werden somit bezeichnet durch $\{C_{j_1^z}^z, C_{j_2^z}^z, \dots, C_{j_m^z}^z\}$ für den z -ten Fountain Decoder.

$$\begin{bmatrix} g_{j_1^z 1} & \cdots & g_{j_1^z k} \\ g_{j_2^z 1} & \cdots & g_{j_2^z k} \\ \vdots & \ddots & \vdots \\ g_{j_m^z 1} & \cdots & g_{j_m^z k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_{j_1^z}^z \\ C_{j_2^z}^z \\ \vdots \\ C_{j_m^z}^z \end{pmatrix} \quad (4)$$

[0067] Somit wird die Linearkombination C_i nicht vollständig verworfen, wenn zumindest ein Fragment während der Übertragung verloren ging. Stattdessen nutzt der Decoder aus, dass Fragmente desselben Typs z zueinander in Beziehung stehen, durch lineare Gleichungen, die in der Erzeugermatrix dargestellt sind.

[0068] Zusätzlich zum Ausnutzen der Tatsache, dass Fragmente desselben Typs z zueinander in Beziehung stehen, führt die vorliegende Erfindung Operationen in diesem Teil, der den b unterschiedlichen Gleichungssystemen gemeinsam ist, lediglich einmal anstatt b mal durch. Somit kann verglichen mit dem oben dargestellten Decodieralgorithmus die Decodierkomplexität weiter verringert werden.

[0069] Ein Fountain Code kann für das Codieren und Decodieren verwendet werden, insbesondere ein Linear Random Fountain Code in Packet Level in einem Galois Feld der Ordnung > 2 .

[0070] Es ist bevorzugt, dass die Wahrscheinlichkeit p_1^b , dass die b Gleichungssysteme zu lösen sind für jeden Fragments typ z , $z = 1, \dots, b$ Gleichungen gemeinsam haben, gegeben ist durch:

$$p_l^b = \binom{k+\Delta}{l} (p_1^b)^l (1-p_1^b)^{k+\Delta-l}$$

wobei $p_1^b = (1-\epsilon)^b$ und ϵ die Frame (Fragment) Auslöschungswahrscheinlichkeit ist, wobei Δ der Overhead auf der TX-Seite ist und $l \in \{0, \dots, k+\Delta\}$.

[0071] Es ist ferner bevorzugt, dass die durchschnittliche Anzahl der gemeinsamen Gleichungen zwischen den b Systemen $\bar{l} = (k + \Delta)p_1^b$ ist.

[0072] In einer bevorzugten Ausführungsform ist die Matrix der Größe $l \times k$, die zusammengesetzt ist aus den gemeinsamen Gleichungen zwischen den b Systemen, bezeichnet als Matrix A. B bezeichnet die Matrix der Größe $h \times k$, die zusammengesetzt ist aus den Gleichungen, die nicht zwischen den b Systemen gemeinsam sind. Die Matrix R, die durch Gauss'sche Elimination zu lösen ist, ist eine Konkatenation der beiden Sub-Matrizen A und B, sodass $R = \begin{bmatrix} A \\ B \end{bmatrix}$. K bezeichnet die Matrix der Größe $(k + \delta)((L/b)/(\log_2 q))$, die sich zusammensetzt aus den $k + \delta$ empfangenen Fragmenten. Somit ist A identisch für jedes der b Gleichungssysteme.

[0073] Während der weiteren Decodierprozedur kann die Matrix R umgeschrieben werden, als eine Konkatenation von fünf Matrizen, L, U, D, C, E, wobei L eine untere Diagonalmatrix der Größe $r \times r$ ist, U eine obere Diagonalmatrix der Größe $r \times r$ ist, D eine Matrix der Größe $r(k - r)$, C ist eine Matrix der Größe $(k - r + \delta)r$, E ist eine Matrix der Größe $(k - r + \delta)(k - r)$. r ist der Rang von A, $r = \text{Rang}(A) \leq l$. Die Matrix L wird dann ausgenullt.

[0074] Die Matrix L kann ausgenullt werden mit der folgenden Anzahl an Operationen an der Matrix, die bezeichnet werden als N_a^L und als N_m^L .

$$N_a^L = \sum_{i=1}^r ((r - i)(k - i))$$

$$N_m^L = \sum_{i=1}^r ((r - i)(1 + k - i))$$

[0075] Dieser Verfahrensschritt an der Matrix L wird für alle b Gleichungssystem einmal ausgeführt und die Matrix C wird ausgenullt. Dies kann getan werden mit der folgenden Anzahl von Operationen, die bezeichnet werden als N_a^C und als N_m^C .

$$N_a^C = \sum_{i=1}^r ((k + \delta - r)(k - i))$$

$$N_m^C = \sum_{i=1}^r ((k + \delta - r)(1 + k - i))$$

[0076] Die Einträge von E werden während des Ausnullens von C modifiziert, sodass E in E' resultiert.

[0077] Ferner wird die folgende Anzahl an Operationen in dem bekannten Term K, die als N_a^K und N_m^K bezeichnet werden, während des Ausnullungsschrittes von L und C ausgeführt.

$$N_a^K = N_m^K = \sum_{i=1}^r \left((k + \delta - i) \left(\frac{\frac{L}{b}}{\log_2 q} \right) \right)$$

[0078] Die Matrix K, nämlich der bekannte Term des Systems nach dem beschriebenen Verfahrensschritten, kann nun als eine Konkatenation der beiden neuen Matrizen K_U der Größe $r \times ((L/b)/\log_2 q)$ und K_L der Größe $(k - r + \delta)$ geschrieben werden als

$$((L/b)/\log_2 q), K = \begin{bmatrix} K_U \\ K_L \end{bmatrix}.$$

[0079] Somit kann die Anzahl der Operationen, die für die beschriebenen Verfahrensschritte notwendig sind und die als N_a^1 und N_m^1 bezeichnet werden, geschrieben werden als

$$N_a^1 = N_a^L + N_a^C + N_a^K$$

$$N_m^1 = N_m^L + N_m^C + N_m^K.$$

[0080] In der weiteren Decodierprozedur ist es bevorzugt, dass Gauss'sche Elimination an der Matrix E' und der Matrix des bekannten Begriffs K_L angewandt wird.

[0081] Die Gleichungen, um die Anzahl der Operationen zu berechnen, die während GE bei einer Matrix der Größe $(k + \delta) \times k$ notwendig sind, sind

$$N_m^{GE}(k, \delta, L, b, q) = \frac{k(k+1)[3(k+\delta)-k-2]}{6} + \left[\frac{k[2(k+\delta)-k-1]}{2} + \frac{k(k+1)}{2} \right] \frac{\frac{L}{b}}{\log_2 q}$$

$$N_a^{GE}(k, \delta, L, b, q) = \frac{k(k-1)[3(k+\delta)-k-1]}{6} + \left[\frac{k[2(k+\delta)-k-1]}{2} + \frac{k(k-1)}{2} \right] \frac{\frac{L}{b}}{\log_2 q}.$$

[0082] Somit kann die Anzahl der Operationen an der Matrix E' der Größe $(k-r+\delta) \times (k-r)$ und den bekannten Ausdruck K_L während der Gauss'schen Elimination erlangt werden durch Auswerten von N_a^{GE} und N_m^{GE} bei $k = k-r$.

[0083] In beiden Gleichungen stellt der Summand die Anzahl der Operationen, die an der Matrix E' während der Vorwärtselimination ausgeführt werden, dar, der zweite Summand stellt die Operationen, die an dem bekannten Ausdruck K_L während der Vorwärtselimination ausgeführt werden, dar und der dritte Summand stellt die Operationen, die während der Rücksubstitution sowohl an der Matrix E' als auch dem bekannten Ausdruck K_L ausgeführt werden, dar.

[0084] Somit ist die Gesamtanzahl der Operationen für die Gauss'sche Elimination N_a^2 und N_m^2

$$N_a^2 = N_a^{E'} + N_a^{K_L} = N_a^{GE}(k=k-r)$$

$$N_m^2 = N_m^{E'} + N_m^{K_L} = N_m^{GE}(k=k-r)$$

wobei $N_a^{E'}$ und $N_m^{E'}$ die Anzahl der Operationen an der Matrix E' sind, während $N_a^{K_L}$ und $N_m^{K_L}$ die Anzahl der Operationen an dem bekannten Ausdruck K_L sind.

[0085] Es ist ferner bevorzugt, dass die Rücksubstitution ausgeführt wird an den verbleibenden Submatrizen U , D und der Matrix des bekannten Ausdrucks K_U .

[0086] Die Gesamtanzahl an Operationen, die für diesen Schritt an der Matrix U , D und an dem bekannten Ausdruck K_U notwendig sind, wird bezeichnet als N_a^3 und N_m^3 und ist gegeben durch die folgenden Gleichungen:

$$N_a^3 = r(k-r) \left(\frac{\frac{L}{b}}{\log_2 q} \right) + \sum_{i=1}^{r-1} \left((r-i) \left(\frac{\frac{L}{b}}{\log_2 q} \right) \right)$$

$$N_m^3 = r(k-r) \left(\frac{\frac{L}{b}}{\log_2 q} \right) + \sum_{i=1}^r \left(i \left(\frac{\frac{L}{b}}{\log_2 q} \right) \right)$$

[0087] Dann kann die Anzahl der Additionen und Multiplikationen im $GF(q)$, die notwendig sind, um die k "virtuellen" Fountain Code Eingangssymbole (Eingangssymbole der Größe L/b Bits) wiederherzustellen, für einen der b Fountain Encoder/Decoder berechnet werden. Sie werden bezeichnet als N_a und N_m ,

$$N_a = N_a^1 + N_a^2 + N_a^3$$

$$N_m = N_m^1 + N_m^2 + N_m^3$$

[0088] Somit kann man die Gesamtanzahl der Operationen, die notwendig sind, um die ganze DATEI zu decodieren (Lösen der b Gleichungssysteme), berechnen.

[0089] Es ist zu beachten, dass die Operationen an der Matrix L , die den b Systemen gemeinsam sind und die bezeichnet werden als N_m^L und N_a^L nur einmal ausgeführt werden müssen.

[0090] Somit wird die Gesamtanzahl der Operationen, um die b Gleichungssysteme mit dem vorgeschlagenen Verfahren zu decodieren, bezeichnet durch $N_a^{\text{TOT-invention}}$ und $N_m^{\text{TOT-invention}}$ und ist gegeben durch die folgenden Gleichungen

$$N_a^{\text{TOT-invention}} = N_a^L + b(N_a^C + N_a^K + N_a^2 + N_a^3)$$

$$N_m^{\text{TOT-invention}} = N_m^L + b(N_m^C + N_m^K + N_m^2 + N_m^3)$$

[0091] Es ist zu beachten, dass die Anzahl der Operationen, die notwendig sind, um die b Gleichungssysteme gemäß dem Stand der Technik zu lösen, ist

$$N_a^{\text{TOT-state-of-the-art}} = bN_a = b \cdot (N_a^L + N_a^C + N_a^K + N_a^2 + N_a^3)$$

$$N_m^{\text{TOT-state-of-the-art}} = bN_m = b \cdot (N_m^L + N_m^C + N_m^K + N_m^2 + N_m^3)$$

[0092] Somit werden, $(b-1)N_m^L$ Multiplikationen und $(b-1)N_a^L$ Additionen eingespart. Somit sind dies die eingesparten Operationen.

[0093] Ferner können wir die Ausdrücke N_m^L und N_a^L durch Verwendung der folgenden Beziehungen entwickeln:

$$\sum_{j=1}^n j = \frac{n(n+1)}{2}$$

$$\sum_{j=1}^n j^2 = \frac{n(n+1)(2n+1)}{6}$$

[0094] Dann wird folgendes erreicht:

$$N_m^L = \frac{r(r+1)[(2r+1)-3(k+1+r)]+6r^2(k+1)}{6}$$

$$N_a^L = \frac{r(r+1)[(2r+1)-3(r+k)]+6kr^2}{6}$$

wo erkannt werden kann, dass die asymptotische Komplexität dieser Terme kubisch ist in $r \sim O(r^3)$. Somit erlaubt es der vorgeschlagene Algorithmus eine beträchtliche Anzahl an Operationen einzusparen, da Operationen in einem kubischen Summanden eingespart werden.

[0095] Es ist ferner bevorzugt, dass Fountain Decoding b mal angewandt wird an kleinen Matrizen der Größe $(k-r+\delta) \times (k-r)$ und nur einmal an einer Matrix der Größe $r \times r$.

[0096] Das vorgeschlagene Decodierverfahren kann verwendet werden mit jeglichen Fountain Codes, z. B. LT, Raptor, LRFC etc. Für LRFC wurden die Vorteile der Erfindung im Hinblick auf die exakte Anzahl der elementaren Operationen bewiesen (ausgehend davon, dass die Dichte der Decodiermatrix gleich 1 ist).

[0097] Im Folgenden werden bevorzugte Ausführungsformen der Erfindung im Kontext der Figuren erläutert.

[0098] [Fig. 1](#) zeigt das Prinzip von Packet Level Fountain Coding,

[0099] [Fig. 2](#) zeigt die Fragmentierung von Fountain Coding Symbolen, gemäß Schema 1, die aus dem Stand der Technik bekannt ist,

[0100] [Fig. 3](#) zeigt Schema 2 ohne Fragmentierung der Fountain Coding Symbole,

[0101] [Fig. 4](#) zeigt ein Beispiel des Performanceverlusts aufgrund von Fragmentierung,

[0102] [Fig. 5](#) zeigt die Performance des Verfahrens, das in DE 10 2011 115 100.5 beschrieben ist, verglichen mit vorher bekannten Verfahren,

[0103] [Fig. 6](#) zeigt eines der b Gleichungssysteme auf der Empfängerseite,

[0104] [Fig. 7](#) bis [Fig. 9](#) zeigen die Matrizen der [Fig. 6](#) nach dem einige Verfahrensschritte des erfindungsgemäßen Decodieralgorithmus angewandt wurden und

[0105] [Fig. 10](#) zeigt die Decodierkomplexität des erfindungsgemäßen Verfahrens verglichen mit dem Stand der Technik.

[0106] [Fig. 1](#) bis [Fig. 5](#) wurden bereits im Kontext des Standes der Technik dargestellt.

[0107] **Fig. 6** zeigt eines der b Gleichungssysteme auf der Empfängerseite. Die linke Seite stellt die Matrix R des Systems dar, während die rechte Seite die Matrix des bekannten Ausdrucks zeigt.

[0108] **Fig. 7** zeigt die Matrix R' des Gleichungssystems nach dem Schritt des Ausnullens der Matrix L . Freie Stellen bezeichnen die Nullen, während Punkte die Einträge der Matrix bezeichnen, die sich von Null unterscheiden. Dieser Teil des Decodieralgorithmus kann lediglich einmal für alle b Systeme angewandt werden.

[0109] Eine detaillierte Beschreibung der Operationen, die an den Matrizen, die in **Fig. 6** bis **Fig. 9** dargestellt sind, angewandt werden, wurde in der Beschreibung der Erfindung in der vorliegenden Patentanmeldung dargestellt.

[0110] **Fig. 8** zeigt die Matrix R'' des Gleichungssystems auf der linken Seite und die Matrix des bekannten Ausdruck auf der rechten Seite, beides auf der Empfängerseite nach den Ausnull-Operationen an der Matrix C .

[0111] **Fig. 9** zeigt die resultierende Matrix des Systems (linke Seite) und seinen bekannten Ausdruck auf der rechten Seite, nachdem Gauss'sche Elimination an der Matrix E' und an der Matrix des bekannten Begriffs K_L angewandt wurden. Der Teil des Systems, der mit Gauss'scher Elimination gelöst wurde, ist nicht dargestellt.

[0112] **Fig. 10** stellt die Decodierkomplexität des erfindungsgemäßen Decodieralgorithmus verglichen mit dem Algorithmus, der in DE 10 2011 115 100.5 beschrieben ist, dar. Ein Beispiel wird angenommen, in dem eine DATEI mit einer Größe von 3 MB übertragen werden soll durch ein Linear Random Fountain Coding Schema über ein Galois Feld der Ordnung 256. Eine MTU von 512 Bytes wird angenommen. Es wird ferner angenommen, dass $b = 4$.

[0113] Somit ist $L = b \cdot \text{MTU} = 2048$ Bytes und $k = 1536$. Eine Frameauslöschungswahrscheinlichkeit ϵ von 0,01 wird angenommen.

[0114] Ferner wird angenommen, dass der Overhead Δ auf der Senderseite gleich 34 ist. Somit werden $34 \times b = 136$ zusätzliche Frames gesendet, um eine Versagenswahrscheinlichkeit von $\sim 10^{-4}$ zu haben. Es wird ferner angenommen, dass jeder der b Fountain Decoder $k + \delta = (k + \Delta) \cdot (1 - \epsilon) = 1554$ Fragmente (Frames). Somit, $\delta = 18$.

[0115] **Fig. 10** stellt die Anzahl der Operationen $N_a + N_m$ gegenüber der Anzahl der gemeinsamen Gleichungen l zwischen den b Gleichungssystemen dar.

[0116] Die durchschnittliche Anzahl der gemeinsamen Gleichungen zwischen den b Systemen ist somit $\bar{l} = 1508$. Somit kann man sagen, dass die Matrix A in **Fig. 6** eine Dimension von 1508×1536 hat, und die Obergrenze der Wahrscheinlichkeit, dass sie einen vollen Rang hat, ist gegeben durch

$$p(\delta', q) = \frac{1}{q-1} q^{-\delta} = \frac{1}{255} 256^{-28} = 1.5 \cdot 10^{-70}.$$

[0117] δ ist in diesem Fall der Overhead von A . Somit kann man vernünftigerweise annehmen, dass $r = \bar{l}$. Jetzt kann die exakte Anzahl der Operationen, die im Stand der Technik notwendig sind, und diejenigen die gemäß der Erfindung notwendig sind, berechnet werden. Ergebnisse sind in Tabelle 1 dargestellt.

[0118] Tabelle 1 zeigt einen Komplexitätsvergleich des Standes der Technik und der Erfindung im Hinblick auf die notwendigen elementaren Operationen für das dargestellte Beispiel. Es ist die große Verringerung der Komplexität zu beachten. Insbesondere erlaubt es die Erfindung 36% der Operationen, die im Stand der Technik notwendig sind, einzusparen.

	Stand der Technik	Erfindung	Komplexitätsverringierung (%) $\left(\frac{N_m^{\text{TOT-state-of-the-art}} - N_m^{\text{TOT-invention}}}{N_m^{\text{TOT-state-of-the-art}}} \right) \times 100$
Anzahl der elementaren Multiplikationen	$N_m^{\text{TOT-state-of-the-art}} = 9.8053e+009$	$N_m^{\text{TOT-invention}} = 6.2806e+009$	36%
Anzahl der elementaren Additionen	$N_a^{\text{TOT-state-of-the-art}} = 9.7973e+009$	$N_a^{\text{TOT-invention}} = 6.2760e+009$	36%

Table 1. Decodierkomplexität. Stand der Technik vs. Erfindung

[0119] Fig. 10 stellt die Gesamtkomplexität des Standes der Technik und die Komplexität der Erfindung gegenüber der Anzahl gemeinsamer Gleichungen l zwischen den b Gleichungssystemen für das Fountain Coding Schema, das in dem Beispiel präsentiert wurde, dar. Es ist zu beachten, dass die Komplexität des Standes der Technik konstant ist, da hier der Vorteil der gemeinsamen Gleichung nicht genutzt wird, während die vorliegende Erfindung dies tut.

[0120] Der vorgeschlagene Decodieralgorithmus kann verwendet werden in allen Arten kommerzieller drahtloser und drahtgebundener Übertragungssysteme, wie z. B. in der Satellitenkommunikation, der Kommunikation via Internet, etc.... Wie gezeigt wurde, erlaubt es das vorgeschlagene Decodierverfahren die Komplexität des Standard Fountain Decodier Algorithmus maßgeblich zu verringern, sodass FC Schemata auch angewandt werden können auf Plattformen mit geringer Rechenleistung.

Patentansprüche

1. Verfahren zum Wiederherstellen von verloren gegangenen und/oder beschädigten Daten, die von einer Sendevorrichtung zu einer Empfängervorrichtung übertragen werden, wobei dieses Verfahren die Schritte aufweist:

Codieren der genannten Daten durch einen Encoder, der mit der Sendevorrichtung verbunden ist, Übertragen der genannten Daten von der Sendevorrichtung zu der Empfängervorrichtung über einen Übertragungskanal und

Decodieren der genannten Daten durch eine Decoder, der mit der Empfängervorrichtung verbunden ist, wobei verloren gegangene und/oder beschädigte Daten während des Decodierens wiederhergestellt werden, wobei eine Erzeugermatrix zum Codieren und Decodieren verwendet wird, wobei Daten vor dem Decodieren in k Eingangssymbole der Länge L aufgeteilt werden, wobei die Ausgangssymbole des Encoders fragmentiert werden, um in die Maximum Transmission Unit (MTU) der Frames im Physical Layer zu passen,

wobei während des Decodierens der Empfänger die empfangenen Fragmente der Ausgangssymbole behandelt, als wären sie Ausgangssymbole von b parallelen Encodern, deren Eingangssymbole k Datenpakete der Größe L/b Bits oder Bytes sind,

wobei die empfangenen Fragmente decodiert werden unter Verwendung von b parallelen Decodern

dadurch gekennzeichnet, dass

die Operationen, die ausgeführt werden an den Gleichungen, die die zu lösenden b Gleichungssysteme gemeinsam haben, einmal anstelle von b mal ausgeführt werden, sodass die Decodierkomplexität verringert wird.

2. Verfahren gemäß Anspruch 1, dadurch gekennzeichnet, dass für $z = 1, \dots, b$ der Empfänger die empfangenen Fragmente behandelt, als wären sie erzeugt gemäß

$$\begin{bmatrix} g_{11} & \cdots & g_{1k} \\ g_{21} & \cdots & g_{2k} \\ \vdots & \ddots & \vdots \\ g_{(k+\Delta)1} & \cdots & g_{(k+\Delta)k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_1^z \\ C_2^z \\ \vdots \\ C_{k+\Delta}^z \end{pmatrix} \quad (3)$$

wobei der linke Teil der Gleichung die Erzeugermatrix ist, P_1^z, \dots, P_k^z die Eingangssymbole der Länge L/b Bits oder Bytes sind und $C_1^z, C_2^z, \dots, C_{k+\Delta}^z$ die Ausgangssymbole sind, und die empfangenen Fragmente decodiert werden unter Verwendung von b parallelen Decodern gemäß:

$$\begin{bmatrix} g_{j_1^z 1} & \cdots & g_{j_1^z k} \\ g_{j_2^z 1} & \cdots & g_{j_2^z k} \\ \vdots & \ddots & \vdots \\ g_{j_m^z 1} & \cdots & g_{j_m^z k} \end{bmatrix} \begin{pmatrix} P_1^z \\ \vdots \\ P_k^z \end{pmatrix} = \begin{pmatrix} C_{j_1^z}^z \\ C_{j_2^z}^z \\ \vdots \\ C_{j_m^z}^z \end{pmatrix} \quad (4)$$

wobei $J^z = \{j_1^z, j_2^z, \dots, j_m^z\}$ das Set von Indices der empfangenen Fragmente für den z -ten Decoder bezeichnen, sodass die empfangenen Fragmente bezeichnet werden, durch $\{C_{j_1^z}^z, C_{j_2^z}^z, \dots, C_{j_m^z}^z\}$ für den z -ten Decoder.

3. Verfahren gemäß einem der Ansprüche 1 bis 2 dadurch gekennzeichnet, dass eine Linearkombination C_j nicht vollständig verworfen wird, wenn wenigstens ein Fragment während der Übertragung verloren ging, sondern der Decoder ausnutzt, dass Fragmente desselben Typs z zueinander in Beziehung stehen durch lineare Gleichungen, die in der Erzeugermatrix dargestellt sind.

4. Verfahren gemäß einem der Ansprüche 1 bis 3 dadurch gekennzeichnet, dass für das Codieren und Decodieren ein Linear Random Fountain Code auf Packet-Level über einem Galois Feld der Ordnung ≥ 2 verwendet wird.

5. Verfahren gemäß einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass die Wahrscheinlichkeit P_1^b , dass die b Gleichungssysteme zu lösen sind für jeden Fragmenttyp z , $z = 1, \dots, b$ Gleichungen gemeinsam haben, gegeben ist durch:

$$P_l^b = \binom{k + \Delta}{l} (P_1^b)^l (1 - P_1^b)^{k + \Delta - l}$$

wobei $P_1^b = (1 - \epsilon)^b$ und ϵ die Frame (Fragment) Auslöschungswahrscheinlichkeit ist, wobei Δ der Overhead auf der TX-Seite ist und $l \in \{0, \dots, k + \Delta\}$

6. Verfahren gemäß Anspruch 5, dadurch gekennzeichnet, dass die durchschnittliche Anzahl der gemeinsamen Gleichungen zwischen den b Systemen $i = (k + \Delta)p_1^b$ ist.

7. Verfahren gemäß einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, dass die Matrix der Größe $l \times k$, die sich aus den gemeinsamen Gleichungen zwischen den b Systemen zusammensetzt, bezeichnet wird als Matrix A ,

B die Matrix der Größe $h \times k$ bezeichnet, die sich zusammensetzt aus den Gleichungen, die zwischen den b Systemen nicht gemeinsam sind,
 und die Matrix R, die durch eine Gauss'sche Elimination zu lösen ist,
 eine Konkatenation der beiden Sub-Matrizen A und B ist, sodass

$$R = \begin{bmatrix} A \\ B \end{bmatrix},$$

wobei die Matrix R umgeschrieben wird als eine Konkatenation von fünf Matrizen L, U, D, C, E, wobei L eine untere Diagonalmatrix der Größe $r \times r$ ist, U eine obere Diagonalmatrix der Größe $r \times r$ ist, D eine Matrix der Größe $r \times (k - r)$, C ist eine Matrix der Größe $(k - r + d) \times r$, E ist eine Matrix der Größe $(k - r + d) \times (k - r)$, r ist der Rang von A, $r = \text{Rang}(A) \leq l$,
 die Matrix L dann ausgenullt wird, wobei dieser Verfahrensschritt an der Matrix L einmal für alle b Systeme ausgeführt wird,
 und die Matrix C ausgenullt wird,
 wobei die Einträge von E während des Ausnullens von C modifiziert werden, sodass aus E E' wird,
 wobei die Matrix K, nämlich der bekannte Ausdruck des Systems nach den beschriebenen Verfahrensschritten nun dargestellt werden kann als eine Konkatenation von zwei neuen Matrizen K_U der Größe $r \times ((L/b)/\log_2 q)$ und K_L der Größe $(k - r + d) \times$

$$((L/b)/\log_2 q), K = \begin{bmatrix} K_U \\ K_L \end{bmatrix}.$$

8. Verfahren gemäß Anspruch 7, dadurch gekennzeichnet, dass Gauss'sche Elimination angewandt wird an der Matrix E' und der Matrix des bekannten Ausdrucks K_L , wobei Rücksubstitution angewandt wird an den verbleibenden Submatrizen U, D, und der Matrix des bekannten Ausdrucks K_U .

9. Verfahren gemäß einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, dass Fountain Decoding b mal an kleineren Matrizen der Größe $(k - r + \delta) \times (k - r)$ angewandt wird und nur einmal an einer Matrix der Größe $r \times r$.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

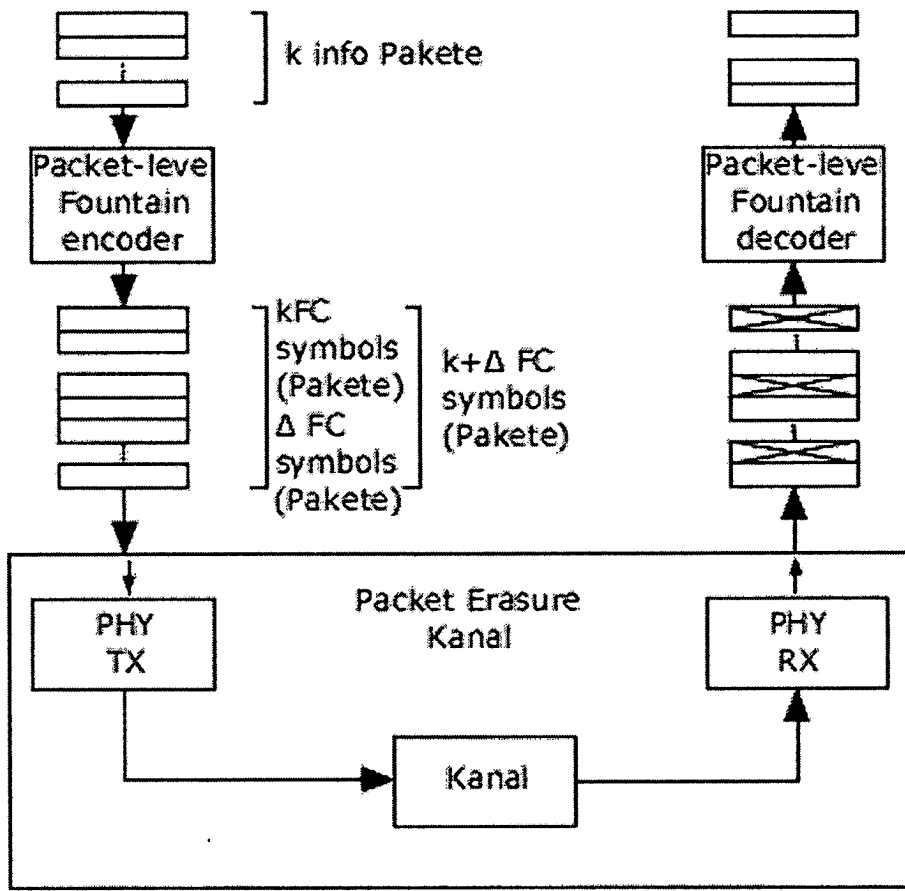


Fig.1 (Stand der Technik)

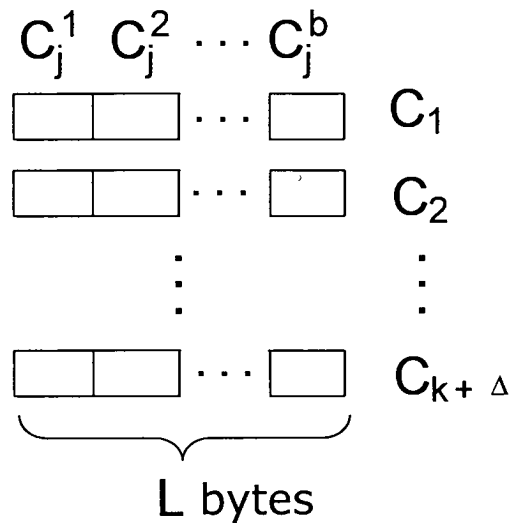


Fig.2 (Stand der Technik)

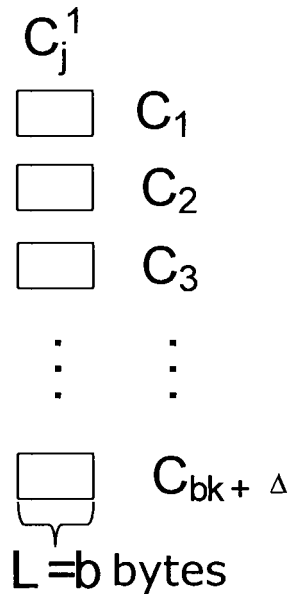


Fig.3 (Stand der Technik)

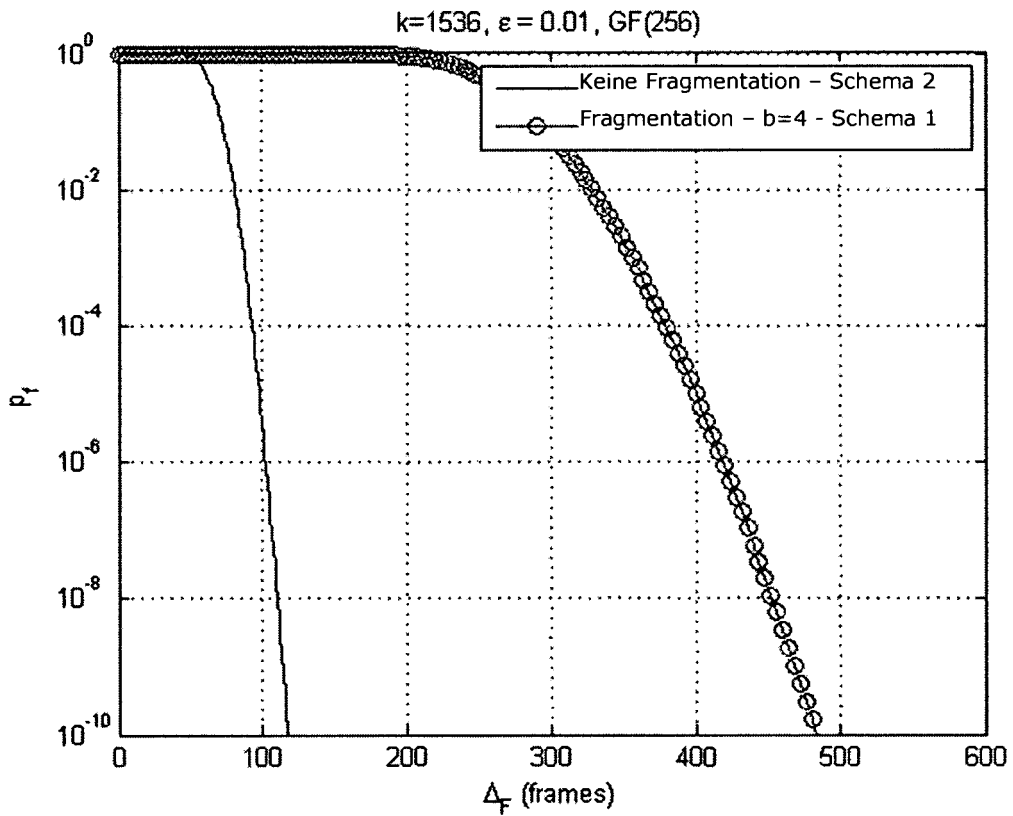


Fig.4 (Stand der Technik)

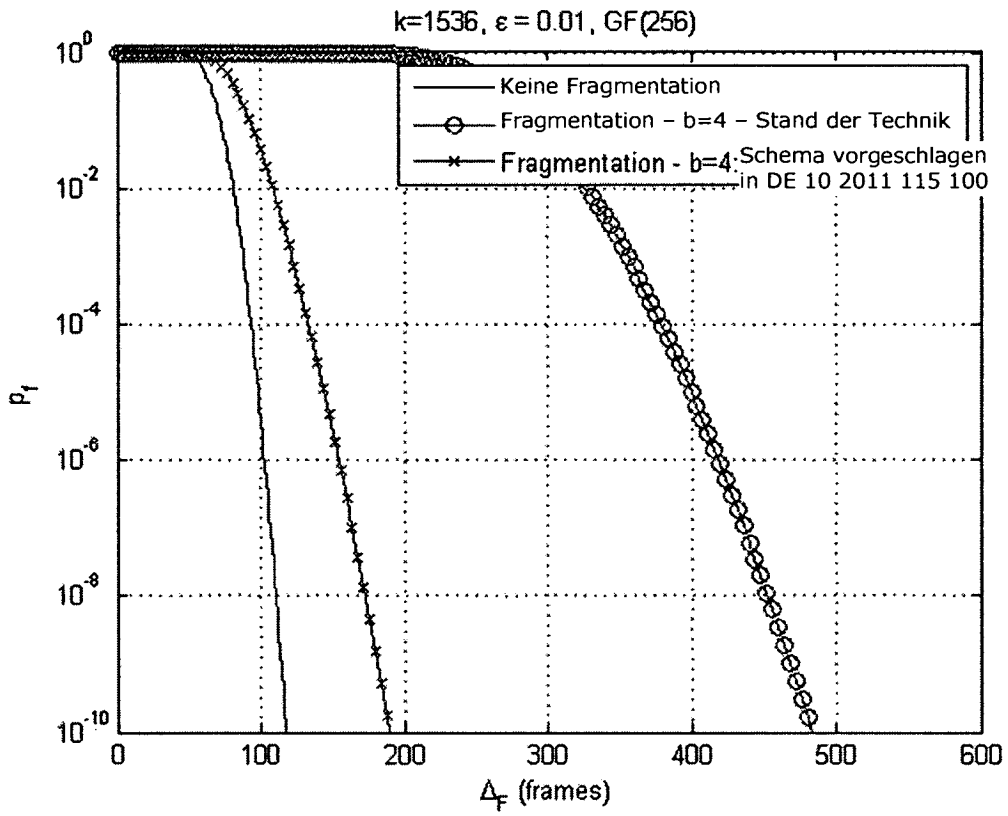


Fig.5 (Stand der Technik)

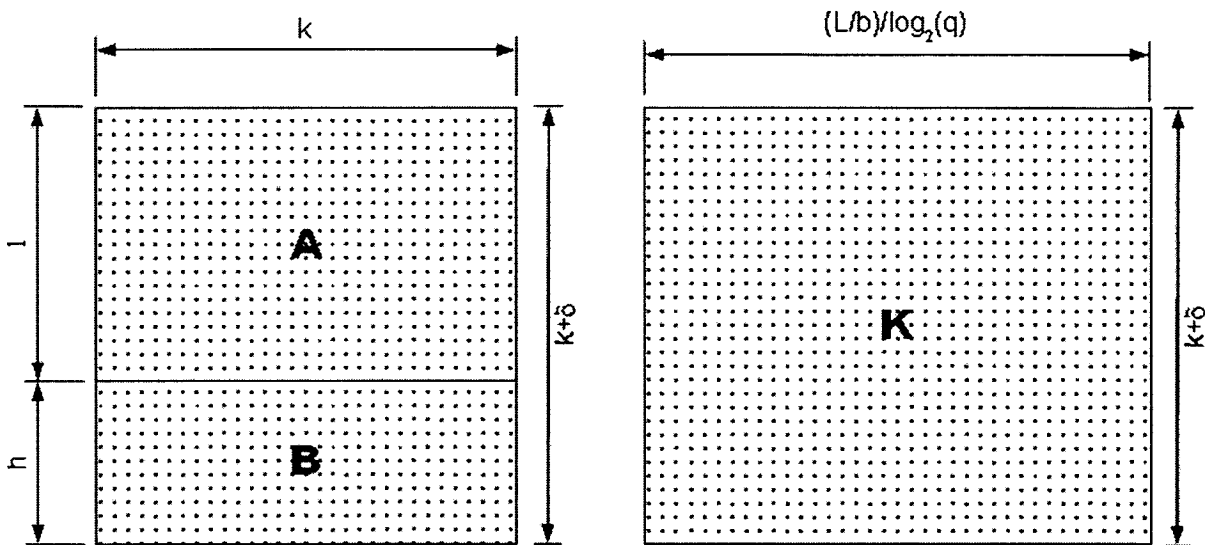


Fig.6

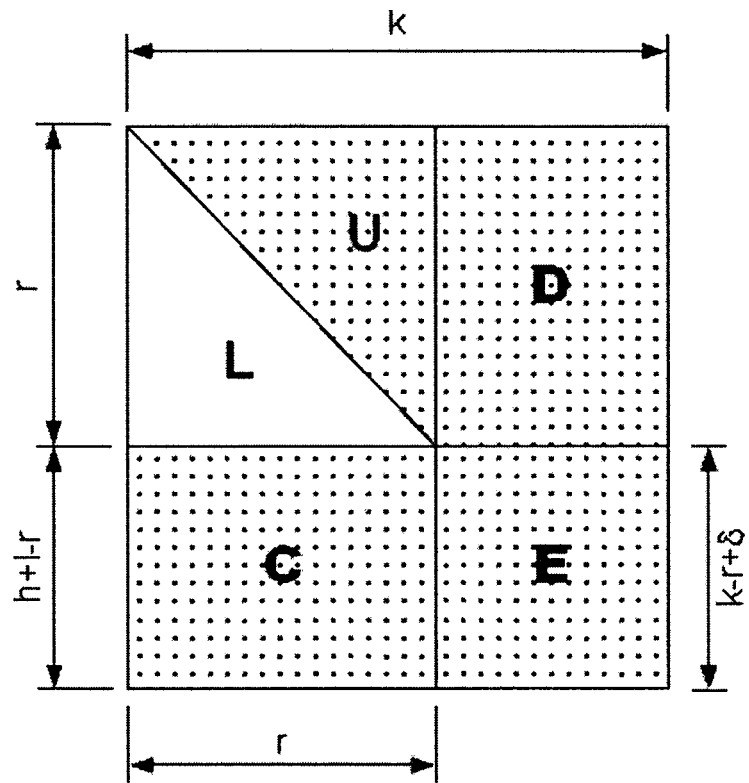


Fig.7

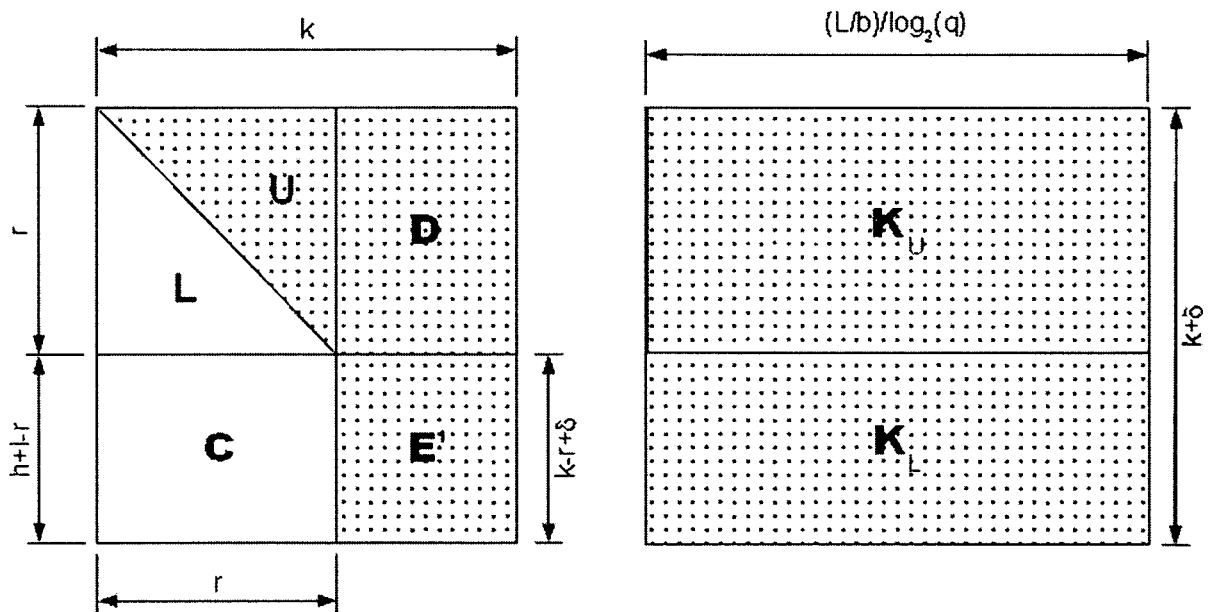


Fig.8

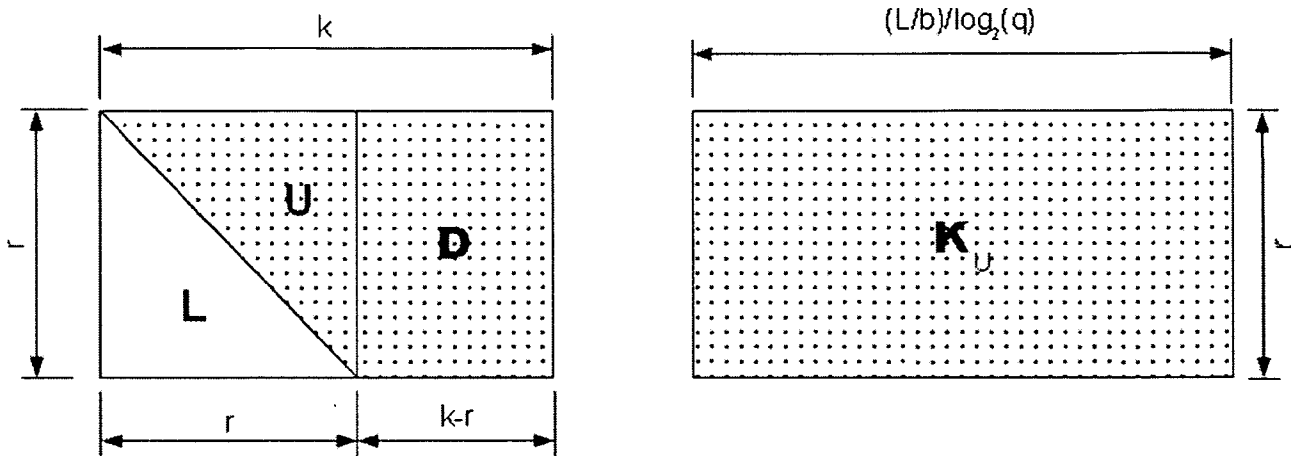


Fig.9

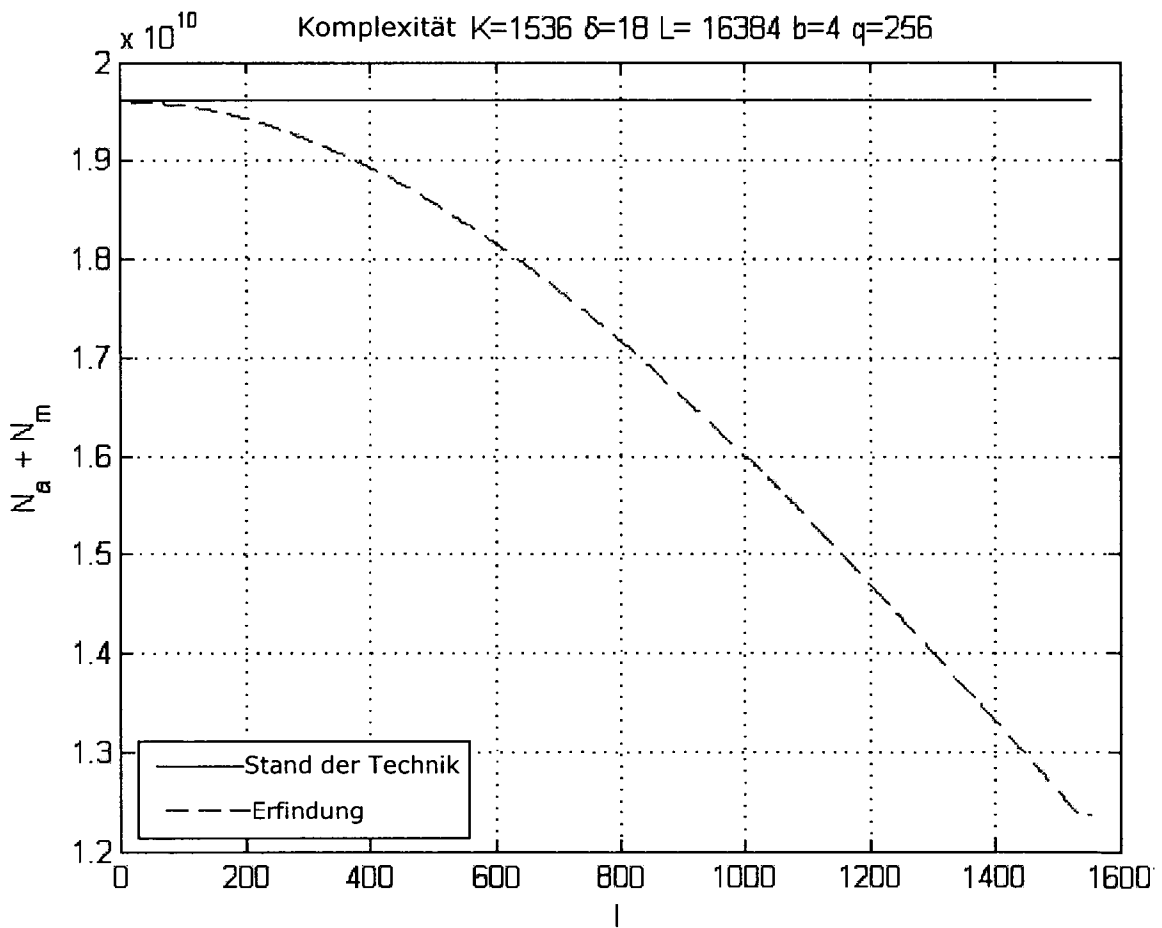


Fig.10