

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro

(43) Internationales Veröffentlichungsdatum  
03. Mai 2018 (03.05.2018)



(10) Internationale Veröffentlichungsnummer  
WO 2018/077771 A2

(51) Internationale Patentklassifikation:  
H04L 1/00 (2006.01)

(21) Internationales Aktenzeichen: PCT/EP2017/076939

(22) Internationales Anmeldedatum:  
23. Oktober 2017 (23.10.2017)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:  
10 2016 220 886.1  
24. Oktober 2016 (24.10.2016) DE

(71) Anmelder: FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V. [DE/DE]; Hansastraße 27c, 80686 München (DE). FRIEDRICH-ALEXANDER-UNIVERSITÄT

ERLANGEN-NUERNBERG [DE/DE]; Schlossplatz 4, 91054 Erlangen (DE).

(72) Erfinder: KILIAN, Gerd; Ungarnstraße 11c, 91056 Erlangen (DE). BERNHARD, Josef; Perschen 14, 92507 Nabburg (DE). ROBERT, Jörg; Erlanger Str. 49a, 91080 Uttenreuth (DE). KNEISSL, Jakob; Alte Reutstr. 46, 90765 Fürth (DE). WECHSLER, Johannes; Egelmühle 1, 91174 Spalt (DE).

(74) Anwalt: HERSINA, Günter et al.; Schoppe, Zimmermann, Stöckeler, Zinkler, Schenk & Partner mbB, Radlkoferstr. 2, 81373 München (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

(54) Title: INTERLEAVING FOR THE TRANSMISSION OF TELEGRAMS WITH A NUMBER OF SUB-PACKETS AND SUCCESSIVE DECODING

(54) Bezeichnung: INTERLEAVING FÜR DIE ÜBERTRAGUNG VON TELEGRAMMEN MIT VARIABLER SUBPAKETANZAHL UND SUKZESSIVER DECODIERUNG

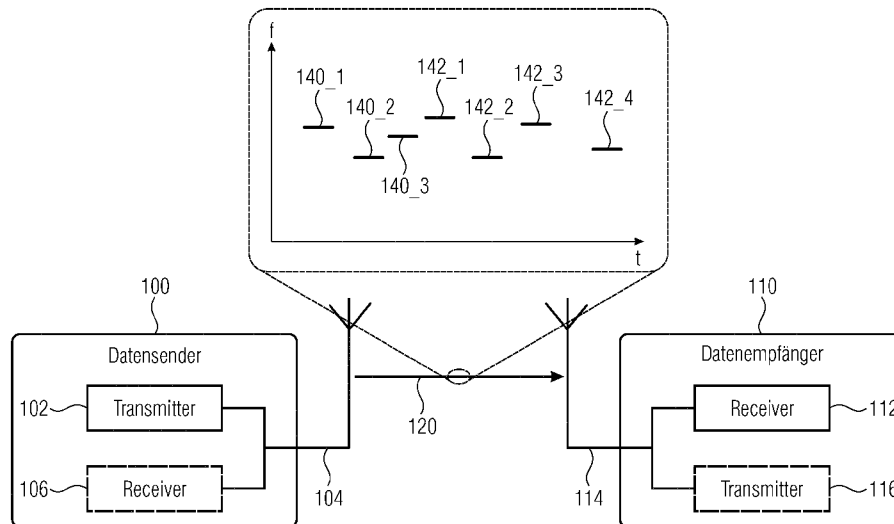


Fig. 1

100 Data transmitter  
110 Data receiver

(57) Abstract: Exemplary embodiments provide a transmission method for the wireless transmission of data in a communications system (e.g. a sensor network or telemetry system). The data comprises core data and additional data, the core data being split in encoded and interleaved form into a plurality of core data sub-packets and the additional data being split in encoded and interleaved form into a plurality of additional data sub-packets, wherein at least some of the core data contained in the core data sub-packets is required for the receipt of the additional data or additional data packets.

(57) Zusammenfassung: Ausführungsbeispiele schaffen ein Übertragungsverfahren zur drahtlosen Übertragung von Daten in einem Kommunikationssystem (z.B. einem Sensornetzwerk oder Telemetriesystem). Die Daten umfassen Kerndaten und Erweiterungsdaten,



WO 2018/077771 A2

HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN,  
KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD,  
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO,  
NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW,  
SA, 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) Bestimmungsstaaten** (soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Veröffentlicht:**

— ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts (Regel 48 Absatz 2 Buchstabe g)

---

wobei die Kerndaten codiert und verschachtelt auf eine Mehrzahl von Kernsubdatenpakete aufgeteilt werden, wobei die Erweiterungsdaten codiert und verschachtelt auf eine Mehrzahl von Erweiterungssubdatenpakete aufgeteilt werden, wobei zumindest ein Teil der in den Kernsubdatenpaketen enthaltenen Kerndaten für einen Empfang der Erweiterungsdaten oder Erweiterungsdatenpakete erforderlich ist.

## Interleaving für die Übertragung von Telegrammen mit variabler Subpaketanzahl und sukzessiver Decodierung

5

### Beschreibung

Ausführungsbeispiele der vorliegenden Erfindung beziehen sich auf einen Datensender zum Senden von Daten. Weitere Ausführungsbeispiele beziehen sich auf einen Datenempfänger zum Empfangen von Daten. Manche Ausführungsbeispiele beziehen sich auf Interleaving für die Übertragung von Telegrammen mit variabler Subpaketanzahl und sukzessiver Decodierung.

In der DE100 2011 082 098 B4 ist ein Verfahren für batteriebetriebene Sender beschrieben, bei dem das Datenpaket in Sendepakete (oder Subdatenpakete) unterteilt wird, die kleiner sind als die eigentliche Information, die übertragen werden soll (so genanntes Telegram Splitting (dt. Telegrammaufteilung)). Telegramme werden dabei auf mehrere Teilpakete (oder Subdatenpakete) aufgeteilt. Ein solches Teilpaket wird als Hop (dt. Sprung) bezeichnet. In einem Hop werden mehrere Informationssymbole übertragen. Die Hops werden auf einer Frequenz oder aber über mehrere Frequenzen verteilt, sog. Frequency Hopping (dt. Frequenzsprung), gesendet. Zwischen den Hops gibt es Pausen, in denen nicht gesendet wird.

Da sich mehrere Funkübertragungen ein Medium zur Übertragung teilen, kann es jederzeit vorkommen, dass ein Subpaket (oder Subdatenpaket) von einer anderen Übertragung so gestört wird, dass es am Empfänger nicht decodiert werden kann. Um diesem Problem entgegenzuwirken, wird eine Kanalcodierung der zu übertragenen Daten durchgeführt, z.B. durch einen Faltungscoder, welcher Redundanz in das Signal gibt, um es auch bei korrumpierten Teilen wieder richtig decodieren zu können.

30

Häufig ist die Leistungsfähigkeit dieser Kanalcodierung davon abhängig, wie viele aufeinanderfolgende Symbole gestört sind. Da durch einen Störer häufig aufeinanderfolgende Symbole gestört werden, werden die Symbole nach der Kanalcodierung so verschachtelt, dass Symbole im ausgesendeten Telegramm, im kanalcodierten Codewort den größtmöglichen Abstand voneinander haben.

35

Das Verschachteln der Symbole (auch Interleaving genannt) hat jedoch zur Folge, dass die Information des Telegramms nur als Ganzes zurückgewonnen werden kann, da die Information über das ganze Telegramm verstreut ist. Dadurch ist es nicht möglich ein Längenfeld zu analysieren, welches es einem erlauben würde, eine variable Anzahl an Subpaketen zu empfangen. Wird z.B. das Längenfeld alleine vorangestellt, ist es nicht teil  
5 der Kanalcodierung und ein Nichtempfang des Subpakets mit der Längeninformation hat den Gesamtverlust des Telegramms zur Folge.

In der DE 10 2011 082 100 A1 wird eine Basisstation mit einer bidirektionalen  
10 Datenübertragung zu einem Knoten beschreiben. Die Basisstation weist eine Einrichtung zum Empfangen eines von dem Knoten mit einer Knoten-Sendefrequenz gesendeten Datenpakets auf, wobei die Knoten-Sendefrequenz von einem Frequenzgeber des Knotens abgeleitet ist. Ferner weist die Basisstation eine Einrichtung zum Ermitteln der Knoten-Sendefrequenz basierend auf dem empfangenen Datenpaket und zum Ermitteln einer  
15 Abweichung des Frequenzgebers des Knotens basierend auf einer Frequenzabweichung zwischen der ermittelten Knoten-Sendefrequenz und einer dem Knoten zugeordneten Soll-Knoten-Sendefrequenz auf. Des Weiteren weist die Basisstation eine Einrichtung zum Senden eines Datenpakets zu dem Knoten mit einer Basisstation-Sendefrequenz auf, wobei die Einrichtung zum Senden des Datenpakets ausgebildet ist, um die Basisstation-  
20 Sendefrequenz basierend auf der ermittelten Abweichung des Frequenzgebers des Knotens einzustellen.

In der WO 2015/128385 A1 wird eine Datensendeordnung beschrieben, die ein Energy-Harvesting-Element als Energiequelle aufweist. Die Datensendeordnung ist dabei  
25 ausgebildet, um Daten unter Verwendung des Telegram Splitting Verfahrens zu senden, wobei ein zum Senden anstehendes Teilpaket in Abhängigkeit von einer von der Energieversorgungseinrichtung bereitstellbaren elektrischen Energiemenge entweder gesendet wird, zwischengespeichert und später gesendet wird, oder verworfen wird.

In der Veröffentlichung [G. Kilian, H. Petkov, R. Psiuk, H. Lieske, F. Beer, J. Robert, and A. Heuberger, "Improved coverage for low-power telemetry systems using telegram splitting," in Proceedings of 2013 European Conference on Smart Objects, Systems and Technologies (SmartSysTech), 2013.] wird eine verbesserte Reichweite für  
35 Niedrigenergie-telemetriesysteme, die das Telegram Splitting Verfahren verwenden, beschrieben.

In der Veröffentlichung [G. Kilian, M. Breiling, H. H. Petkov, H. Lieske, F. Beer, J. Robert, and A. Heuberger, "Increasing Transmission Reliability for Telemetry Systems Using Telegram Splitting," IEEE Transactions on Communications, vol. 63, no. 3, pp. 949–961, Mar. 2015.] wird eine verbesserte Übertragungssicherheit für  
5 Niedrigenergie-telemetriesysteme, die das Telegram Splitting Verfahren verwenden, beschrieben.

In der Veröffentlichung [Sam Dolinar, Dariush Divsalar, and Fabrizio Pollara, "Turbo Code Performance as a Function of Code Block Size", 1998 IEEE International Symposium on  
10 Information Theory] wird die Leistungsfähigkeit von Turbo Codes in Abhängigkeit von einer Blockgröße beschrieben.

Der vorliegenden Erfindung liegt somit die Aufgabe zugrunde, eine Übertragungssicherheit bei der Übertragung von Daten variabler Länge zu erhöhen.

15

Diese Aufgabe wird durch die unabhängigen Patentansprüche gelöst.

Vorteilhafte Weiterbildungen finden sich in den abhängigen Patentansprüchen.

20 Ausführungsbeispiele schaffen einen Datensender, der ausgebildet ist, um Kerndaten zu codieren und verschachtelt auf eine Mehrzahl von Kernsubdatenpakete aufzuteilen, wobei der Datensender ausgebildet ist, um Erweiterungsdaten zu codieren und verschachtelt auf eine Mehrzahl von Erweiterungssubdatenpakete aufzuteilen, wobei zumindest ein Teil der in den Kernsubdatenpaketen enthaltenen Kerndaten für einen Empfang der Erweiterungsdaten  
25 oder Erweiterungsdatenpakete erforderlich ist.

Weitere Ausführungsbeispiele schaffen einen Datenempfänger, der ausgebildet ist, um Kernsubdatenpakete und Erweiterungssubdatenpakete zu empfangen, wobei die Kernsubdatenpakete Kerndaten enthalten, die über die Kernsubdatenpakete verschachtelt  
30 verteilt sind, und wobei die Erweiterungssubdatenpakete Erweiterungsdaten enthalten, die über die Erweiterungssubdatenpakete verschachtelt verteilt sind, wobei der Datenempfänger ausgebildet ist, um zumindest einen Teil der codierten Kerndaten zu decodieren, um eine Information bezüglich der Erweiterungsdatenpakete zu erhalten, und wobei der Datenempfänger ausgebildet ist, um die Erweiterungsdatenpakete unter Verwendung der  
35 Information zu empfangen.

Gemäß dem Konzept der vorliegenden Erfindung werden zur Übertragung von Daten variabler Länge Kernsubdatenpakete und Erweiterungssubdatenpakete verwendet, wobei codierte Kerndaten der Daten variabler Länge auf die Kernsubdatenpakete verschachtelt aufgeteilt werden, um die Übertragungssicherheit der codierten Kerndaten zu erhöhen, und  
5 wobei codierte Erweiterungsdaten der Daten variabler Länge auf Erweiterungssubdatenpakete und optional auf die Kernsubdatenpakete verschachtelt aufgeteilt werden, um die Übertragungssicherheit der codierten Erweiterungsdaten zu erhöhen, wobei die codierten Kerndaten eine Information über die codierten Erweiterungsdaten oder Erweiterungssubdatenpakete enthalten.

10

Weitere Ausführungsbeispiele schaffen ein Verfahren zum Senden von Kerndaten und Erweiterungsdaten. Das Verfahren umfasst einen Schritt des Codierens von Kerndaten, um codierte Kerndaten zu erhalten. Ferner umfasst das Verfahren einen Schritt des Verschachtelns und Aufteilens der codierten Kerndaten auf eine Mehrzahl von  
15 Kernsubdatenpakete. Ferner umfasst das Verfahren einen Schritt des Codierens von Erweiterungsdaten, um codierte Erweiterungsdaten zu erhalten. Ferner umfasst das Verfahren einen Schritt des Verschachtelns und Aufteilens der codierten Erweiterungsdaten auf eine Mehrzahl von Erweiterungsdatenpaketen. Ferner umfasst das Verfahren einen Schritt des Sendens der Kernsubdatenpakete und Erweiterungsdatenpakete.

20

Weitere Ausführungsbeispiele schaffen ein Verfahren zum Empfangen von Kerndaten und Erweiterungsdaten. Das Verfahren umfasst einen Schritt des Empfangens von Kernsubdatenpaketen und Erweiterungssubdatenpaketen, wobei die Kernsubdatenpakete Kerndaten enthalten, die über die Kernsubdatenpakete verschachtelt verteilt sind, und wobei  
25 die Erweiterungssubdatenpakete Erweiterungsdaten enthalten, die über die Erweiterungssubdatenpakete verschachtelt verteilt sind. Ferner umfasst das Verfahren einen Schritt des Decodierens zumindest eines Teils der codierten Kerndaten, um eine Information bezüglich der Erweiterungsdatenpakete zu erhalten, wobei die Erweiterungsdatenpakete unter Verwendung der Information empfangen werden.

30

Weitere Ausführungsbeispiele schaffen ein Übertragungsverfahren zur drahtlosen Übertragung von Daten in einem Kommunikationssystem (z.B. einem Sensornetzwerk oder Telemetriesystem). Die Daten umfassen Kerndaten und Erweiterungsdaten, wobei die Kerndaten codiert und verschachtelt auf eine Mehrzahl von Kernsubdatenpakete aufgeteilt  
35 werden, wobei die Erweiterungsdaten codiert und verschachtelt auf eine Mehrzahl von Erweiterungssubdatenpakete aufgeteilt werden, wobei zumindest ein Teil der in den

Kernsubdatenpaketen enthaltenen Kerndaten für einen Empfang der Erweiterungsdaten oder Erweiterungsdatenpakete erforderlich ist.

Im Folgenden werden bevorzugte Ausführungsbeispiele des Datensenders beschrieben.

5

Bei Ausführungsbeispielen ist der Datensender ausgebildet, um keine codierten Kerndaten auf die Erweiterungssubdatenpakete aufzuteilen. Mit anderen Worten, die Erweiterungssubdatenpakete enthalten keine codierten Kerndaten.

10 Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die codierten Kerndaten auf die Kernsubdatenpakete derart aufzuteilen, dass auch bei Übertragungsverlust eines oder mehrerer der Kernsubdatenpakete eine empfängerseitige Decodierung der Kerndaten basierend auf den anderen Kernsubdatenpaketen möglich ist.

15 Beispielsweise können die die codierten Kerndaten nacheinander auf die Kernsubdatenpakete aufgeteilt werden, so dass der Verlust eines der Kernsubdatenpakete nicht zum Totalverlust der Decodiermöglichkeit führt. Hierzu können beispielsweise unmittelbar aufeinander folgende Symbole der Kerndaten auf unmittelbar aufeinander folgende Kernsubdatenpakete aufgeteilt werden.

20

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die codierten Kerndaten derart auf die Kernsubdatenpakete aufzuteilen, dass ein zeitlicher Abstand der codierten Kerndaten in Bezug auf eine Einflusslänge eines für die Codierung der Kerndaten verwendeten Codes (Kanalcodes) erhöht (oder sogar maximiert) ist.

25

Beispielsweise kann der Datensender ausgebildet sein, um Symbole der codierten Kerndaten derart auf die Kernsubdatenpakete aufzuteilen, dass ein zeitlicher Abstand der Symbole in Bezug auf eine Einflusslänge eines für die Codierung der Kerndaten verwendeten Codes (Kanalcodes) erhöht (oder sogar maximiert) ist.

30

Die Leistungsfähigkeit der Kanalcodierung kann davon abhängig sein, wie viele aufeinanderfolgende Symbole gestört sind. Da durch einen Störer häufig aufeinanderfolgende Symbole gestört werden, können die Symbole nach der Kanalcodierung so verschachtelt werden, dass diese in den ausgesendeten Kernsubdatenpaketen den  
35 größtmöglichen Abstand voneinander haben.

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um, falls eine Länge der Kerndaten zur Auffüllung der Kernsubdatenpakete nicht ausreicht, die Erweiterungsdaten (oder einen Teil der Erweiterungsdaten) verschachtelt auf die Kernsubdatenpakete aufzuteilen, um die Kernsubdatenpakete aufzufüllen.

5

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um bei der Auffüllung der Kernsubdatenpakete die Erweiterungsdaten (oder einen Teil der Erweiterungsdaten) derart auf die Kernsubdatenpakete und Erweiterungssubdatenpakete aufzuteilen, dass ein Abstand der codierten Erweiterungsdaten in Bezug auf eine Einflusslänge eines für die Codierung der Erweiterungsdaten verwendeten Codes (Kanalcodes) erhöht (oder sogar maximiert) ist.

10

Beispielsweise kann der Datensender ausgebildet sein, um Symbole der codierten Erweiterungsdaten derart auf die Kernsubdatenpakete aufzuteilen, dass ein zeitlicher Abstand der Symbole in Bezug auf eine Einflusslänge eines für die Codierung der Kerndaten verwendeten Codes (Kanalcodes) erhöht (oder sogar maximiert) ist.

15

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um bei der Auffüllung der Kernsubdatenpakete die Erweiterungsdaten derart auf die Kernsubdatenpakete und Erweiterungssubdatenpakete aufzuteilen, dass die Kernsubdatenpakete und Erweiterungssubdatenpakete gleichmäßig aufgefüllt werden.

20

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um bei der Auffüllung der Kernsubdatenpakete die Erweiterungsdaten derart auf die Kernsubdatenpakete und Erweiterungssubdatenpakete aufzuteilen, dass die Kernsubdatenpakete und Erweiterungssubdatenpakete ungleichmäßig aufgefüllt werden.

25

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die codierten Kerndaten auf eine feste oder vorgegebene Anzahl von Kernsubdatenpakete aufzuteilen.

30

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um eine Anzahl von Erweiterungssubdatenpaketen in Abhängigkeit von einer Länge der Erweiterungsdaten anzupassen.

35

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die Kerndaten und die Erweiterungsdaten gemeinsam zu codieren.

Dabei können die Kerndaten und die Erweiterungsdaten gemeinsam so codiert werden, dass eine Decodierung der codierten Kerndaten zumindest einen Teil der Kerndaten liefert.

5 Beispielsweise kann bei manchen Kanalcodes die Leistungsfähigkeit erhöht werden, wenn die Eingabelänge der Daten steigt, weshalb die Kerndaten und die Erweiterungsdaten gemeinsam codiert werden können. Hierbei ist jedoch zu beachten, dass die Kerndaten und die Erweiterungsdaten gemeinsam so codiert werden, dass eine Decodierung der Kerndaten oder zumindest eines Teils der Kerndaten auch ohne die Erweiterungsdaten möglich ist.

10 Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die Kerndaten und die Erweiterungsdaten unabhängig voneinander zu codieren.

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um die uncodierten Kerndaten mit Erweiterungsdaten zu füllen, so dass die Erweiterungsdaten zeitlich vor den  
15 Kerndaten angeordnet sind und die Sicherheit bei der Decodierung der Kerndaten erhöht ist.

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um zumindest einen Anteil der Kernsubdatenpakete mit Synchronisationsdaten zu versehen.

20 Dabei kann der Datensender ausgebildet sein, um in den jeweiligen Kernsubdatenpaketen die Kerndaten zeitlich benachbart zu den Synchronisationsdaten anzuordnen.

Ferner kann der Datensender ausgebildet sein, um in zeitlich (unmittelbar) aufeinanderfolgenden Kernsubdatenpaketen die Kerndaten abwechselnd vor und nach den  
25 Synchronisationsdaten anzuordnen.

Des Weiteren kann der Datensender ausgebildet sein, um in den jeweiligen Kernsubdatenpaketen die Synchronisationsdaten zeitlich so anzuordnen, dass diese unmittelbar benachbart zu den Erweiterungsdaten und unmittelbar benachbart zu den  
30 Kerndaten angeordnet sind.

Bei Ausführungsbeispielen kann der Datensender ausgebildet sein, um reine Synchronisationssubdatenpakete auszusenden.

35 Der Datensender kann hierbei ausgebildet sein, die Kernsubdatenpakete und die Synchronisationssubdatenpakete so auszusenden, dass die Kernsubdatenpakete und die Synchronisationssubdatenpakete zeitlich benachbart zueinander angeordnet sind.

Beispielsweise können die Synchronisationssubdatenpakete zwischen den Kernsubdatenpaketen ausgesendet werden.

5 Im Folgenden werden bevorzugte Ausführungsbeispiele des Datenempfängers beschrieben.

Bei Ausführungsbeispielen kann dem Datenempfänger die Anzahl an Kernsubdatenpaketen bekannt sein.

10 Bei Ausführungsbeispielen kann die Information bezüglich der Erweiterungsdatenpakete, die in den Kernsubdatenpaketen enthalten ist, eine Anzahl an Erweiterungssubdatenpaketen sein.

Bei Ausführungsbeispielen können die codierten Kerndaten auf die Kernsubdatenpakete  
15 derart aufgeteilt sein, dass auch bei Übertragungsverlust eines oder mehrerer der Kernsubdatenpakete eine empfängerseitige Decodierung der Kerndaten basierend auf den anderen Kernsubdatenpaketen möglich ist. Der Datenempfänger kann dabei ausgebildet sein, um zumindest einen Anteil der Kernsubdatenpakete zu empfangen und zu decodieren, um die Kerndaten zu erhalten.

20 Beispielsweise können die codierten Kerndaten nacheinander auf die Kernsubdatenpakete aufgeteilt sein, so dass der Verlust eines der Kernsubdatenpakete nicht zum Totalverlust der Decodiermöglichkeit führt. Hierzu können beispielsweise unmittelbar aufeinander folgende Symbole der Kerndaten auf unmittelbar aufeinander folgende Kernsubdatenpakete aufgeteilt  
25 sein.

Bei Ausführungsbeispielen kann zumindest ein Anteil der Kernsubdatenpakete mit Synchronisationsdaten versehen sein, wobei der Datenempfänger ausgebildet sein kann, um die Kernsubdatenpakete basierend auf zumindest einem Anteil der Synchronisationsdaten in  
30 einem Empfangsdatenstrom zu detektieren.

Bei Ausführungsbeispielen kann der Datenempfänger ausgebildet sein, um reine Synchronisationssubdatenpakete zu empfangen, und um die Kernsubdatenpakete basierend auf zumindest einem Anteil der Synchronisationssubdatenpakete in einem  
35 Empfangsdatenstrom zu detektieren.

Bei Ausführungsbeispielen kann der Datenempfänger ausgebildet sein, um zumindest einen Teil der decodierten Kerndaten zu reencodieren, um reencodierte Kerndaten zu erhalten, und um zumindest einen Teil der codierten Erweiterungsdaten unter Verwendung der reencodierten Kerndaten zu decodieren.

5

Bei Ausführungsbeispielen kann der Datenempfänger ausgebildet sein, um einen ersten Teil der codierten Erweiterungsdaten zu decodieren und zu reencodieren, um einen ersten Teil von reencodierten Erweiterungsdaten zu erhalten, und um einen zweiten Teil der codierten Erweiterungsdaten unter Verwendung des ersten Teils von reencodierten Erweiterungsdaten zu decodieren.

10

Ausführungsbeispiele der vorliegenden Erfindung werden bezugnehmend auf die beiliegenden Figuren näher erläutert. Es zeigen:

- 15 Fig. 1 ein schematisches Blockschaltbild eines Systems mit einem Datensender und einem Datenempfänger, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;
- Fig. 2 eine schematische Ansicht einer Kanalcodierung und Symbolzuordnung von  
20 Daten;
- Fig. 3 in einem Diagramm eine Aufteilung der Kerndaten und Erweiterungsdaten in Kernsubdatenpakete und Erweiterungssubdatenpakete;
- 25 Fig. 4 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete;
- Fig. 5 in einem Diagramm eine Aufteilung der Kernsymbole und  
30 Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete nach einem ersten Auffüllungszwischenergebnis;
- Fig. 6 in einem Diagramm eine Aufteilung der Kernsymbole und  
35 Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete nach einem zweiten Auffüllungszwischenergebnis;

- Fig. 7 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete nach einem dritten Auffüllungszwischenergebnis;  
5
- Fig. 8 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete nach einem vierten Auffüllungszwischenergebnis;  
10
- Fig. 9 eine schematische Ansicht einer zyklischen Verschiebung des Gesamtworts nach der Kanalcodierung, so dass Symbole des Erweiterungsworts vor dem Kernwort angeordnet sind;  
15
- Fig. 10 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den Subdatenpaketen den jeweiligen Kernsymbolen bzw. Erweiterungssymbolen Synchronisationssymbole vorangestellt sind;  
20
- Fig. 11 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in der Mitte der jeweiligen Subdatenpakete zwischen den jeweiligen Kernsymbolen bzw. Erweiterungssymbolen Synchronisationssymbole vorhanden sind;  
25
- Fig. 12 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei zwischen den Kernsubdatenpakete Synchronisationssubdatenpakete mit Synchronisationssymbolen angeordnet sind;  
30
- Fig. 13 zeigt in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den jeweiligen Subdatenpaketen zwei  
35

voneinander beabstandete (Teil-)Synchronisationssymbolsequenzen vorhanden sind;

- 5 Fig. 14 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den jeweiligen Subdatenpaketen zwei
- 10 Fig. 15 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den jeweiligen Subdatenpaketen in der
- 15 Fig. 16 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den jeweiligen Subdatenpaketen in der
- 20 Fig. 17 in einem Diagramm eine Aufteilung der Kernsymbole und Erweiterungssymbole in Kernsubdatenpakete und Erweiterungssubdatenpakete, wobei in den jeweiligen Subdatenpaketen in der
- 25 Fig. 18 zeigt in einem Diagramm eine Aufteilung der Kernsymbole in Kernsubdatenpakete, wobei in den jeweiligen Kernsubdatenpaketen in der
- 30 Fig. 19 in einem Diagramm eine Aufteilung der Kernsymbole in Kernsubdatenpakete, wobei in den jeweiligen Kernsubdatenpaketen in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen vorhanden sind, wobei die Kernsymbole in aufeinander folgenden Kernsubdatenpaketen abwechselnd vor und nach den (Teil-)Synchronisationssymbolsequenzen aufgeteilt sind;
- 35 Fig. 20 ein Flussdiagramm eines Verfahrens zum Senden von Kerndaten und Erweiterungsdaten

Fig. 21 ein Flussdiagramm eines Verfahrens zum Empfangen von Kerndaten und Erweiterungsdaten, gemäß einem Ausführungsbeispiel.

In der nachfolgenden Beschreibung der Ausführungsbeispiele der vorliegenden Erfindung werden in den Figuren gleiche oder gleichwirkende Elemente mit dem gleichen Bezugszeichen versehen, so dass deren Beschreibungen in den unterschiedlichen Ausführungsbeispielen untereinander austauschbar ist.

Fig. 1 zeigt ein schematisches Blockschaltbild eines Systems mit einem Datensender 100 zum Senden von Daten 120 und einem Datenempfänger 110 zum Empfangen von Daten 120, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. Die Daten 120 können Kerndaten und Erweiterungsdaten aufweisen.

Der Datensender 100 ist ausgebildet, um die Kerndaten zu codieren und verschachtelt auf eine Mehrzahl von Kernsubdatenpakete 140\_1 bis 140\_n aufzuteilen, und um die Erweiterungsdaten zu codieren und verschachtelt auf eine Mehrzahl von Erweiterungssubdatenpakete 142\_1 bis 142\_m aufzuteilen, wobei zumindest ein Teil der in den Kernsubdatenpaketen 140\_1 bis 140\_n enthaltenen Kerndaten für einen Empfang der Erweiterungsdatenpakete erforderlich ist.

Der Datenempfänger 110 ist ausgebildet, um die Kernsubdatenpakete 140\_1 bis 140\_n und die Erweiterungssubdatenpakete 142\_1 bis 142\_m zu empfangen, wobei die Kernsubdatenpakete 140\_1 bis 140\_n die Kerndaten enthalten, die über die Kernsubdatenpakete 140\_1 bis 140\_n verschachtelt verteilt sind, und wobei die Erweiterungssubdatenpakete 142\_1 bis 142\_m die Erweiterungsdaten enthalten, die über die Erweiterungssubdatenpakete 142\_1 bis 142\_m verschachtelt verteilt sind. Der Datenempfänger 110 ist ferner ausgebildet, um zumindest einen Teil der codierten Kerndaten zu decodieren, um eine Information bezüglich der Erweiterungsdatenpakete 142\_1 bis 142\_m zu erhalten, und wobei der Datenempfänger 110 ausgebildet ist, um die Erweiterungsdatenpakete 142\_1 bis 142\_m unter Verwendung der Information zu empfangen.

Bei Ausführungsbeispielen können die Kerndaten auf n Kernsubdatenpakete 140\_1 bis 140\_n aufgeteilt sein, wobei n eine natürliche Zahl größer gleich zwei ist,  $n \geq 2$ . Die Erweiterungsdaten können auf m Erweiterungssubdatenpakete 142\_1 bis 142\_m aufgeteilt sein. Die Daten 120 können also unter Verwendung von  $s=n+m$  Subdatenpaketen (Kernsubdatenpakete + Erweiterungssubdatenpakete) übertragen werden, wobei  $s_{\min}=n$  die

Mindestanzahl (= Anzahl n an Kernsubdatenpaketen 140\_1 bis 140\_n) an zu übertragenden Subdatenpaketen ist.

Bei Ausführungsbeispielen können die Erweiterungsdaten sowohl auf die  
5 Kernsubdatenpakete 140\_1 bis 140\_n als auch auf die Erweiterungssubdatenpakete 142\_1 bis 142\_m verschachtelt aufgeteilt sein. Die Kernsubdatenpakete 140\_1 bis 140\_n können also sowohl Kerndaten als auch einen Teil der Erweiterungsdaten enthalten.

Bei Ausführungsbeispielen werden die Kerndaten nicht auf die Erweiterungssubdatenpakete  
10 142\_1 bis 142\_m aufgeteilt. Die Erweiterungssubdatenpakete 142\_1 bis 142\_m enthalten also keine Kerndaten.

Bei Ausführungsbeispielen können die Subdatenpakete (Kernsubdatenpakete +  
Erweiterungssubdatenpakete) mit einem zeitlichen Abstand übertragen werden, so dass  
15 zwischen den Subdatenpaketen Sendepausen vorhanden sind.

Bei Ausführungsbeispielen können die Subdatenpakete (Kernsubdatenpakete und/oder  
Erweiterungssubdatenpakete) unter Verwendung eines Zeitsprungmusters und/oder  
20 Frequenzsprungmusters übertragen werden.

Bei Ausführungsbeispielen kann das Frequenzsprungmuster eine Abfolge von  
Sendefrequenzen oder Sendefrequenzsprüngen angeben, mit denen die Subdatenpakete zu  
senden sind.

Beispielsweise kann ein erstes Subdatenpaket mit einer ersten Sendefrequenz (oder in  
einem ersten Frequenzkanal) und ein zweites Subdatenpaket mit einer zweiten  
Sendefrequenz (oder in einem zweiten Frequenzkanal) gesendet werden, wobei die erste  
Sendefrequenz und die zweite Sendefrequenz unterschiedlich sind. Das  
Frequenzsprungmuster kann dabei die erste Sendefrequenz und die zweite Sendefrequenz  
30 definieren (oder vorgeben, oder angeben). Alternativ kann das Frequenzsprungmuster die  
erste Sendefrequenz und einen Frequenzabstand (Sendefrequenzsprung) zwischen der  
ersten Sendefrequenz und der zweiten Sendefrequenz angeben. Natürlich kann das  
Frequenzsprungmuster auch nur den Frequenzabstand (Sendefrequenzsprung) zwischen  
der ersten Sendefrequenz und der zweiten Sendefrequenz angeben.

35 Bei Ausführungsbeispielen kann das Zeitsprungmuster eine Abfolge von Sendezeitpunkten  
oder Sendezeitabständen angeben, mit denen die Subdatenpakete zu senden sind.

Beispielsweise kann ein erstes Subdatenpaket zu einem ersten Sendezeitpunkt (oder in einem ersten Sendezeitschlitz) und ein zweites Subdatenpaket zu einem zweiten Sendezeitpunkt (oder in einem zweiten Sendezeitschlitz) gesendet werden, wobei der erste Sendezeitpunkt und der zweite Sendezeitpunkt unterschiedlich sind. Das Zeitsprungmuster kann dabei den ersten Sendezeitpunkt und den zweiten Sendezeitpunkt definieren (oder vorgeben, oder angeben). Alternativ kann das Zeitsprungmuster den ersten Sendezeitpunkt und einen zeitlichen Abstand zwischen dem ersten Sendezeitpunkt und dem zweiten Sendezeitpunkt angeben. Natürlich kann das Zeitsprungmuster auch nur den zeitlichen Abstand zwischen dem ersten Zeitpunkt und dem zweiten Sendezeitpunkt angeben.

Ein Zeit- und Frequenzsprungmuster kann die Kombination aus einem Frequenzsprungmuster und einem Zeitsprungmuster sein, d.h. eine Abfolge von Sendezeitpunkten oder Sendezeitabständen mit denen die Subdatenpakete übertragen werden, wobei den Sendezeitpunkten (oder Sendezeitabständen) Sendefrequenzen (oder Sendefrequenzsprünge) zugeordnet sind.

Bei Ausführungsbeispielen können die Kernsubdatenpakete 140\_1 bis 140\_n und die Erweiterungssubdatenpakete 142\_1 bis 142\_m mit separaten Zeitsprungmustern und/oder Frequenzsprungmustern übertragen werden.

Bei Ausführungsbeispielen kann der Datensender 100 eine Sendeeinrichtung (Transmitter) 102 aufweisen, die ausgebildet ist, um die Daten 120 zu senden. Die Sendeeinrichtung 102 kann mit einer Antenne 104 des Datensenders 100 verbunden sein. Der Datensender 100 kann ferner eine Empfangseinrichtung (Receiver) 106 aufweisen, die ausgebildet ist, um Daten zu empfangen. Die Empfangseinrichtung kann mit der Antenne 104 oder einer weiteren (separaten) Antenne des Datensenders 100 verbunden sein. Der Datensender 100 kann auch eine kombinierte Sendeempfangseinrichtung (Transceiver) aufweisen.

Der Datenempfänger 110 kann eine Empfangseinrichtung (Receiver) 116 aufweisen, die ausgebildet ist, um die Daten 120 zu empfangen. Die Empfangseinrichtung 116 kann mit einer Antenne 114 des Datenempfängers 110 verbunden sein. Ferner kann der Datenempfänger 110 eine Sendeeinrichtung (Transmitter) 112 aufweisen, die ausgebildet ist, um Daten zu senden. Die Sendeeinrichtung 112 kann mit der Antenne 114 oder einer weiteren (separaten) Antenne des Datenempfängers 110 verbunden sein. Der Datenempfänger 110 kann auch eine kombinierte Sendeempfangseinrichtung (Transceiver) aufweisen.

Bei Ausführungsbeispielen kann der Datensender 100 ein Sensorknoten sein, während der Datenempfänger 110 eine Basisstation sein kann. Natürlich ist es auch möglich, dass der Datensender 100 eine Basisstation ist, während der Datenempfänger 110 ein Sensorknoten ist. Ferner ist es möglich, dass sowohl der Datensender 100 als auch der Datenempfänger 110 Sensorknoten sind. Des Weiteren ist es möglich, dass sowohl der Datensender 100 als auch der Datenempfänger 110 Basisstationen sind.

Bei Ausführungsbeispielen lassen sich die zu sendenden Daten in zwei Teile unterteilen, die sogenannte Kerninformation, welche schon vor Erhalt des Gesamtpakets verarbeitet werden kann und die Erweiterungsinformation. Werden die Informationen kanalcodiert und zugeordnet (oder gemappt) so ergeben sie das Kernwort, beziehungsweise das Erweiterungswort, wie dies in Fig. 2 gezeigt ist.

Im Detail zeigt Fig. 2 eine schematische Ansicht einer Kanalcodierung und Symbolzuordnung von Daten. Wie in Fig. 2 zu erkennen ist, können die Daten 120 eine Kerninformation (Kerndaten) 122 und eine Erweiterungsinformation (Erweiterungsdaten) 124 aufweisen. Die Kanalcodierung und Symbolzuordnung der Kerninformation 122 kann ein Kernwort 130 mit den Symbolen  $k_0$  bis  $k_K$  (Kernsymbole 136) ergeben. Die Kanalcodierung und Symbolzuordnung der Erweiterungsinformation 124 kann ein Erweiterungswort 132 mit den Symbolen  $e_0$  bis  $e_E$  (Erweiterungssymbole 138) ergeben. Das Kernwort 130 und das Erweiterungswort 132 können ein Gesamtwort 134 bilden. Mit anderen Worten, Fig. 2 zeigt die Bildung des Gesamtwortes 134 aus Kerninformation 122 und Erweiterungsinformation 124.

Das gezielte Verschachteln der Kernwörter 130 wie in hierin beschrieben, ist im Besonderen dann wichtig, wenn es im Kanal zu Störungen kommt. Hierbei ist es von Vorteil, wenn das Gesamtdatenpaket, das übertragen werden soll, in kleinere Teilpakete, sogenannte Subpakete unterteilt wird (siehe DE 10 2011 082 098 B4). Ist der zeitliche Abstand zwischen Subpaketen lange genug im Vergleich zu den im Kanal auftretenden Störern, so ist die Wahrscheinlichkeit hoch, dass nur ein einziges Subpaket gestört wird. Sind die Daten nun passend verschachtelt, hat der Verlust der Information eines Subpakets noch nicht den Verlust der Information zur Folge. Um dies zu erreichen kann eine Kanalcodierung auf die Information angewendet werden bevor diese in Subpakete aufgeteilt wird.

35

Nach der Kanalcodierung können die Symbole auf das Sendalphabet des Übertragungsverfahrens gemappt (oder zugeordnet) werden. Die dadurch entstehenden Wörter werden als Kernwort bzw. Erweiterungswort bezeichnet.

- 5 Diese Wörter bilden dann ein Gesamtwort 134, welches dann in Subpakete unterteilt wird. Das Kernwort kann nun so in das Übertragungswort verschachtelt werden, dass dieses möglichst früh am Empfänger ausgewertet werden kann, wie dies in Fig. 3 gezeigt ist.

10 Im Detail zeigt Fig. 3 in einem Diagramm eine Aufteilung der Kerndaten 122 und Erweiterungsdaten 124 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_5. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_5), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen  
15 Subdatenpaketen beschreibt.

Wie in Fig. 3 zu erkennen ist, kann das Kernwort 130 auf die Kerndatenpakete 140\_1 bis 140\_4 aufgeteilt werden. Im Detail können die Symbole 136 des Kernworts 130 verschachtelt auf die Kernsubdatenpakete 140\_1 bis 140\_4 aufgeteilt werden. Genauso  
20 kann das Erweiterungswort 132 auf die Erweiterungssubdatenpakete 142\_1 bis 142\_5 aufgeteilt werden. Im Detail können die Symbole 138 des Erweiterungsworts 132 verschachtelt auf die Erweiterungssubdatenpakete 142\_1 bis 142\_5 aufgeteilt werden.

Wie in Fig. 3 ferner zu erkennen ist, kann das Erweiterungswort 132 (oder im Detail die Erweiterungssymbole 138) sowohl auf die Kerndatenpakete 140\_1 bis 140\_4 als auch auf  
25 die Erweiterungssubdatenpakete 142\_1 bis 142\_5 aufgeteilt werden. Dies ist insbesondere dann möglich, wenn die Kernsubdatenpakete 140\_1 bis 140\_4 durch das Kernwort 130 (oder im Detail die Kernsymbole 136) nicht vollständig aufgefüllt werden. In diesem Fall kann das Erweiterungswort 132 sowohl auf die Kerndatenpakete 140\_1 bis 140\_4 als auch auf die  
30 Erweiterungssubdatenpakete 142\_1 bis 142\_5 aufgeteilt werden.

Mit anderen Worten, Fig. 3 zeigt dass Kernsubpakete 140\_1 bis 140\_4 Kernsymbole 136 enthalten, Erweiterungssubpakete 142\_1 bis 142\_5 enthalten keine Kernsymbole 136. Die Subpakete lassen sich demnach in zwei verschiedene Kategorien einteilen. Erstens,  
35 Kernsubpakete 140\_1 bis 140\_4. Dies sind Subpakete 140\_1 bis 140\_4, die Symbole 136 des Kernwortes 130 enthalten. Zweitens, Erweiterungssubpakete 142\_1 bis 142\_5. Dies sind Subpakete 142\_1 bis 142\_5, die keine Symbole des Kernwortes 130 enthalten. Eine

traditionelle Übertragung (also ohne Subpakete), kann erreicht werden, wenn die Subpakete ohne zeitlichen Abstand übertragen werden.

Im Folgenden werden detaillierte Ausführungsbeispiele des oben vorgestellten Übertragungsverfahrens, das durch den Datensender 100 und den Datenempfänger 110 durchgeführt werden kann, näher erläutert.

#### Erstes detailliertes Ausführungsbeispiel: Interleaving der Kerndaten auf Kernsubpakete

10 Das Kernwort 132, welches z.B. wichtige Seiteninformationen zum Empfang enthalten kann, kann hierbei komplett in die sogenannten Kernsubdatenpaketen 140\_1 bis 140\_n verschachtelt werden, welche vom Empfänger 110 vor dem abgeschlossenen Empfang der Erweiterungssubpakete 142\_1 bis 142\_m ausgewertet werden können. Die Anzahl n der Kernsubpakete 140\_1 bis 140\_n muss dem Empfänger 110 bekannt sein.

15

Hierbei ist es wichtig, dass auch der Verlust eines oder mehrerer der Kernsubpakete 140\_1 bis 140\_n, z.B. durch eine Störung, kompensiert werden kann. Dafür können die Symbole der Kerndaten 122 gleichmäßig über das Kernsubpaket verteilt werden, da der Verlust eines größeren zusammenhängenden Blocks in den Daten für viele Kanalcodierungsverfahren zum schnellen Versagen führt.

20

Die einfachste Vorgehensweise hierfür ist, die Symbole nacheinander in die Subpakete der Kernsequenz (Kernsubdatenpakete 140\_1 bis 140\_n) zu verschachteln. Ist die Anzahl n der Kernsubpakete 140\_1 bis 140\_n erreicht, so beginnt die Zuteilung wieder von vorne. Die Anzahl der Subpakete in der Kernsequenz ist für alle Telegramme variabler Länge identisch und gegeben, d.h. fest vorgegeben.

25

Fig. 4 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e19 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_5. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_5), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

35

Wie in Fig. 4 zu erkennen ist können die Kernsymbole k0 bis k15 auf die Kernsubdatenpakete 140\_1 bis 140\_4 nacheinander verschachtelt aufgeteilt werden,

während die Erweiterungssymbole  $e_0$  bis  $e_{19}$  auf die Erweiterungssubdatenpakete 142\_1 bis 142\_m nacheinander verschachtelt verteilt werden. Im Detail können zunächst die Kernsymbole  $k_0$  bis  $k_{15}$  auf die Kernsubdatenpakete 140\_1 bis 140\_4 verschachtelt aufgeteilt werden, so dass der Verlust eines der Kernsubdatenpakete 140\_1 bis 140\_3 nicht

5 zum Totalverlust der Decodiermöglichkeit führt. Beispielsweise kann das erste Kernsubdatenpaket 140\_1 die Kernsymbole  $k_0$ ,  $k_4$ ,  $k_8$  und  $k_{12}$ , das zweite Kernsubdatenpaket 140\_2 die Kernsymbole  $k_1$ ,  $k_5$ ,  $k_9$ ,  $k_{13}$ , das dritte Kernsubdatenpaket 140\_3 die Kernsymbole  $k_2$ ,  $k_6$ ,  $k_{10}$  und  $k_{14}$ , und das dritte Kernsubdatenpaket 140\_4 die Kernsymbole  $k_3$ ,  $k_7$ ,  $k_{11}$  und  $k_{15}$  enthält.

10

Mit anderen Worten, Fig. 4 zeigt eine beispielhafte Aufteilung der Kern- und Erweiterungssymbole mit  $n=S_{\min}=4$  Kernsubpaketen,  $m=S-S_{\min}=5$  Erweiterungssubpakete 142\_1 bis 142\_m mit einer Symbolanzahl pro Subpaket von  $N_S=4$ .

15 Bei dem in Fig. 4 gezeigten Beispiel werden beispielhaft  $N_K=16$  Kernsymbole  $k_0$  bis  $k_{15}$  auf  $n=S_{\min}=4$  Kernsubdatenpakete 140\_1 bis 140\_4 verschachtelt.

Zum Beispiel können die Symbole wie folgt verschachtelt werden:

0. Symbol in 0. Kernsubpaket

20 1. Symbol in 1. Kernsubpaket

...

3. Symbol in 3. Kernsubpaket

4. Symbol in 0. Kernsubpaket

5. Symbol in 1. Kernsubpaket

25 ...

15. Symbol in 4. Kernsubpaket

Dieses ist nur ein Beispiel, prinzipiell kann auf beliebige Art verschachtelt werden, einige Wege der Verschachtelung werden allerdings zu schlechterer Leistungsfähigkeit bei Verlust

30 von Subpaketen führen.

Bei Ausführungsbeispielen kann die Information in Subpakete verschachtelt werden, so dass das Kernwort getrennt und vor dem Erweiterungswort empfangen werden kann.

35 Bei Ausführungsbeispielen können die Symbole innerhalb der Kern- bzw. Erweiterungssubpakete derart verschachtelt werden, dass der Verlust eines oder mehrerer Kernsequenzsubpakets nicht zum Totalverlust der Decodiermöglichkeit führt.

Zweites detailliertes Ausführungsbeispiel: Auffüllen der Kernsubpakete mit Erweiterungssymbolen für variable Subpaketanzahl

5 Üblicherweise ist die Anzahl der Symbole, die in den Kernsubpaketen 140\_1 bis 140\_n ausgesendet werden können, größer als die Anzahl der Symbole im Kernwort 130, so dass die Kernsubpakete 140\_1 bis 140\_n mit Symbolen aus dem Erweiterungswort 132 aufgefüllt werden können. Damit dies möglichst gleichmäßig geschieht, kann das Verhältnis von Kernsymbolen im Kernsubpaket ( $N_K$ ) und Erweiterungssymbolen im Kernsubpaket ( $N_S - N_K$ )  
 10 vor dem verschachteln festgelegt werden. Dieses Verhältnis  $V = \frac{N_K}{N_S - N_K}$  kann ein ganzzahliger Teiler der Anzahl der Kernsubpakete ( $S_{min}$ ) sein und die Anzahl der Kernsymbole in einem Subpaket kann größer gleich der Anzahl an Erweiterungssymbolen in einem Subpaket sein, also  $\frac{N_K}{N_S} \geq 1$ .

15 Werden das Kernwort 130 und das Erweiterungswort 132 nun auf das Telegramm aufgeteilt, so wird das Kernwort in die ersten freien Datensymbole der Kernsubpakete aufgeteilt.

Die ersten  $\frac{S_{min}}{V}$  Symbole des Erweiterungswortes 132 werden dann in die Kernsubpakete 140\_1 bis 140\_n verteilt. Diese Zuordnung kann gleichmäßig passieren, so dass nicht zwei  
 20 dieser Symbole im gleichen Subpaket platziert werden.

Hierfür ergeben sich verschiedene Möglichkeiten. Erstens, die Symbole können im Abstand  $V$  in die Kernsubpakete platziert werden. Beim ersten Durchlauf startet dieser Vorgang bei Subpaket 0 und geht in  $V$  Schritten vorwärts. Das zweite Symbol kann also in Subpaket  $V$   
 25 platziert werden, usw. Zweitens, die Symbole können in  $V$  Blöcken in die Kernsubpakete platziert werden. Die  $\frac{S_{min}}{V}$  Symbole belegen  $\frac{S_{min}}{V}$  hintereinanderliegende Kernsubpakete, überschreitet der Kernsubpaketindex  $S_{min}$ , so wird wieder bei Kernsubpaket 0 begonnen.

Die nächsten  $S - S_{min}$  Erweiterungssymbole können dann in die Subpakete der  
 30 Erweiterungssequenz verteilt werden, ohne zusätzlichen Abstand. Starten also bei Subpaket  $S_{min}$  und enden bei Subpaket  $S$ .

Wurden  $S - S_{min} + \frac{S_{min}}{V}$  Erweiterungssymbole verteilt, so startet man beim Platzieren der nächsten  $\frac{S_{min}}{V}$  Erweiterungssymbole bei Subpaket 1 (142\_2) und geht wieder in  $V$  Schritten  
 35 vorwärts. Das nächste Symbol wird also in Subpaket  $V + 1$  platziert usw, bis die  $\frac{S_{min}}{V}$

Symbole platziert sind. Die Platzierung der Erweiterungssymbole innerhalb der Erweiterungssequenz bleibt wie beim ersten Schritt.

Ist jedem Subpaket der Kernsequenz ein Erweiterungssymbol zugeordnet worden, so  
5 beginnt die nächste Zuordnungsrunde wieder bei Subpaket 0 (140\_1) und die Methode wird fortgeführt bis zum Ende der Erweiterungssymbole.

Fig. 5 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und  
Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und  
10 Erweiterungssubdatenpakete 142\_1 bis 142\_3 nach einem ersten  
Auffüllungszwischenergebnis. Dabei beschreibt die Abszisse eine zeitliche Anordnung der  
Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete  
142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole  
(Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

15  
Wie in Fig. 5 zu erkennen ist, können zunächst die Kernsymbole k0 bis k15 auf die  
Kernsubdatenpakete 140\_1 bis 140\_4 verschachtelt aufgeteilt werden, so dass der Verlust  
eines oder mehrerer der Kernsubdatenpakete 140\_1 bis 140\_3 nicht zum Totalverlust der  
Decodiermöglichkeit führt. Im Detail, kann das erste Kernsubdatenpaket 140\_1 die  
20 Kernsymbole k0, k4, k8 und k12, das zweite Kernsubdatenpaket 140\_2 die Kernsymbole k1,  
k5, k9, k13, das dritte Kernsubdatenpaket 140\_3 die Kernsymbole k2, k6, k10 und k14, und  
das dritte Kerndatenpaket 140\_4 die Kernsymbole k3, k7, k11 und k15 enthält.

Anschließend können die Erweiterungssymbole e0 bis e39 sowohl auf die  
25 Kernsubdatenpakete 140\_1 bis 140\_4 als auch auf die Erweiterungssubdatenpakete  
verschachtelt verteilt werden, so dass die Kernsubdatenpakete 140\_1 bis 140\_4 und die  
Erweiterungssubdatenpakete 142\_1 bis 142\_3 gleichmäßig aufgefüllt werden, und so dass  
der Verlust eines der Subdatenpakete nicht zum Totalverlust der Decodiermöglichkeit führt.

30 Ziel ist es, dass das erste Kernsubdatenpaket 140\_1 die Erweiterungssymbole e0, e10, e20  
und e30, das zweite Kernsubdatenpaket 140\_2 die Erweiterungssymbole e5, e15, e25 und  
e35, das dritte Kernsubdatenpaket 140\_3 die Erweiterungssymbole e1, e11, e21 und e31,  
das vierte Kernsubdatenpaket 140\_4 die Erweiterungssymbole e6, e16, e26 und e36, das  
erste Erweiterungssubdatenpaket 142\_1 die Erweiterungssymbole e2, e7, e12, e17, e22,  
35 e27, e32, und e37, das zweite Erweiterungssubdatenpaket 142\_2 die Erweiterungssymbole  
e3, e8, e13, e18, e23, e28, e33 und e38, und das dritte Erweiterungssubdatenpaket 142\_3  
die Erweiterungssymbole e4, e9, e14, e19, e24, e29, e34 und e39 enthalten.

Hierzu kann zunächst das Erweiterungssymbol e0 auf das erste Kernsubdatenpaket 140\_1, das Erweiterungssymbol e1 auf das dritte Kernsubdatenpaket 140\_3, das Erweiterungssymbol e2 auf das erste Erweiterungssubdatenpaket 142\_1, das Erweiterungssymbol e3 auf das zweite Erweiterungssubdatenpaket 142\_2, und das das Erweiterungssymbol e3 auf das dritte Erweiterungssubdatenpaket 142\_3 aufgeteilt werden.

Mit anderen Worten, Fig. 5 zeigt ein Platzieren der Erweiterungssymbole e0 und e1 mit Abstand  $V=2$  in den Kernsubpaketen startend bei Kernsubpaket 0, Symbole e2, e3 und e4 werden in die Erweiterungssubpakete verschachtelt.

Fig. 6 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, nach einem zweiten Auffüllungszwischenergebnis. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 6 zu erkennen ist, kann nun das Erweiterungssymbol e5 auf das zweite Kernsubdatenpaket 140\_2, das Erweiterungssymbol e6 auf das vierte Kernsubdatenpaket 140\_4, das Erweiterungssymbol e7 auf das erste Erweiterungssubdatenpaket 142\_1, das Erweiterungssymbol e8 auf das zweite Erweiterungssubdatenpaket 142\_2, und das das Erweiterungssymbol e9 auf das dritte Erweiterungssubdatenpaket 142\_3 aufgeteilt werden.

Mit anderen Worten, Fig. 6 zeigt ein Platzieren der Erweiterungssymbole e5 und e6 mit Abstand  $V=2$  in den Kernsubpaketen startend bei Kernsubpaket 1, Symbole e7, e8 und e9 werden in die Erweiterungssubpakete verschachtelt.

Fig. 7 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, nach einem dritten Auffüllungszwischenergebnis. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 7 zu erkennen ist, kann nun das Erweiterungssymbol e10 auf das erste Kernsubdatenpaket 140\_1, das Erweiterungssymbol e11 auf das dritte Kernsubdatenpaket 140\_3, das Erweiterungssymbol e12 auf das erste Erweiterungssubdatenpaket 142\_1, das Erweiterungssymbol e13 auf das zweite Erweiterungssubdatenpaket 142\_2, und das das  
5 Erweiterungssymbol e14 auf das dritte Erweiterungssubdatenpaket 142\_3 aufgeteilt werden.

Mit anderen Worten, Fig. 7 zeigt ein Platzieren der Erweiterungssymbole e10 und e11 mit Abstand  $V=2$  in den Kernsubpaketen startend bei Kernsubpaket 0, Symbole e12, e13 und e14 werden in die Erweiterungssubpakete verschachtelt.

10

Fig. 8 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, nach einem vierten Auffüllungszwischenergebnis. Dabei beschreibt die Abszisse eine zeitliche Anordnung der  
15 Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 8 zu erkennen ist, kann nun das Erweiterungssymbol e15 auf das zweite  
20 Kernsubdatenpaket 140\_2, das Erweiterungssymbol e16 auf das vierte Kernsubdatenpaket 140\_4, das Erweiterungssymbol e17 auf das erste Erweiterungssubdatenpaket 142\_1, das Erweiterungssymbol e18 auf das zweite Erweiterungssubdatenpaket 142\_2, und das das Erweiterungssymbol e19 auf das dritte Erweiterungssubdatenpaket 142\_3 aufgeteilt werden.

25 Mit anderen Worten, Fig. 8 zeigt ein Platzieren der Erweiterungssymbole e15 und e16 mit Abstand  $V=2$  in den Kernsubpaketen startend bei Kernsubpaket 1, Symbole e17, e18 und e19 werden in die Erweiterungssubpakete verschachtelt.

Die oben genannten Schritte können dann fortgeführt werden bis alle Symbole verteilt sind.

30

Bei Ausführungsbeispielen können die Erweiterungssymbole in Kernsubpakete verschachtelt werden, sodass die Gesamtzahl der Subpakete gleichmäßig aufgefüllt wird.

Ausführungsbeispiele ermöglichen bzw. unterstützen eine variable Anzahl an Subpaketen.

35

Drittes detailliertes Ausführungsbeispiel: Gemeinsame Kanalcodierung von Kerninformation und Erweiterungsinformation

Bei manchen Kanalcodes (z. B. Turbo-Codes) lässt sich die Performance erhöhen, wenn die Eingabelänge der Daten (Block-Größe) steigt. Diese Tatsache wird ausführlich in [Sam Dolinar, Dariush Divsalar, and Fabrizio Pollara, "Turbo Code Performance as a Function of Code Block Size", 1998 IEEE International Symposium on Information Theory] beschrieben.

Des Weiteren wird in [G. Kilian, M. Breiling, H. H. Petkov, H. Lieske, F. Beer, J. Robert, and A. Heuberger, "Increasing Transmission Reliability for Telemetry Systems Using Telegram Splitting," IEEE Transactions on Communications, vol. 63, no. 3, pp. 949–961, Mar. 2015.] gezeigt, dass die Dekodierwahrscheinlichkeit im Interferenzkanal mit zunehmender Anzahl an Teilpaketen steigt. Werden zusätzlich zum Kernwort nur wenig Symbole im Erweiterungswort übertragen, müssen entsprechend wenige Subpakete an die Kernsubpakete angefügt werden.

Durch die beiden oben beschriebenen Effekte sinkt dann die Dekodierwahrscheinlichkeit bei einer kurzen Erweiterungsinformation bzw. Kerninformation stark ab, wenn die Kanalcodierung für beide getrennt durchgeführt wird.

Um diesen beiden Effekten entgegenzuwirken, kann Kanalcode über die komplette Information, also Kerninformation und Erweiterungsinformation, berechnet werden.

Bei Ausführungsbeispielen kann eine Übertragungssicherheit von kurzen Kern- oder Erweiterungsinformationen durch gemeinsame Kanalcodierung erhöht werden.

Viertes detailliertes Ausführungsbeispiel: Sicherung der Decodierbarkeit der Kerninformation durch Einfügen von Erweiterungssymbolen

Für Graphen basierte Decoder ohne definierte Anfangs- bzw. Endzustände, z.B. Viterbi Faltungsdecoder (mit Tailbiting), sind die ersten empfangenen Symbole nur mit geringer Sicherheit decodierbar, da noch keine ausreichende Datenmenge zum Bestimmen des besten Pfads vorhanden ist. Die Kerninformation ist jedoch essentiell für den Empfang und sollte mit einer hohen Sicherheit decodiert werden können. Um dieses Problem zu lösen, können der Kerninformation entsprechend so viele Daten der Erweiterungsinformation vorangestellt werden (in der Regel (typischerweise) die Länge der Einflusslänge des Codes), dass eine sichere Decodierung der Kerninformation möglich ist.

Dies ist beispielsweise dadurch zu erreichen, dass nach der Kanalcodierung das Gesamtwort mit vorangestelltem Kernwortteil zyklisch um die Anzahl der Symbole verschoben wird, welche aus dem Erweiterungsteil stammen, wie dies in Fig. 9 gezeigt ist.

- 5 Im Detail zeigt Fig. 9 eine schematische Ansicht einer zyklischen Verschiebung des Gesamtworts nach der Kanalcodierung, so dass Z Symbole des Erweiterungsworts 132 vor dem Kernwort 130 angeordnet sind.

Wie in Fig. 9 zu erkennen ist, umfasst das Kernwort 130 nach der Kanalcodierung K+1  
10 Symbole  $e_0$  bis  $e_K$ , wobei das Erweiterungswort 132 nach der Kanalcodierung E+1 Symbole  $e_0$  bis  $e_E$  umfasst. Von den E+1 Symbole  $e_0$  bis  $e_E$  des Erweiterungsworts 132 können Z Symbole entnommen werden und den K+1 Symbolen  $e_0$  bis  $e_K$  des Kernworts 130 vorangestellt werden. Beispielsweise können die letzten Z Symbole  $e_Z$  bis  $e_E$  des Erweiterungsworts 132 entnommen werden und den K+1 Symbolen  $e_0$  bis  $e_K$  des Kernworts  
15 130 vorangestellt werden.

Mit anderen Worten, Fig. 9 zeigt ein Beispiel zum Einfügen von Z Erweiterungssymbolen (hellblau) in den Bereich der Kernsymbole durch zyklisches verschieben nach der Kanalcodierung.

20

Eine andere Möglichkeit wäre jede andere Kombination von Symbolen an den Anfang zu schieben oder direkt Erweiterungsinformation in den Kern übernehmen bevor Kanalcodiert wird.

- 25 Bei Ausführungsbeispielen kann eine gemeinsame Kanalcodierung von Kern und Erweiterung (vgl. drittes detailliertes Ausführungsbeispiel) erfolgen.

Bei Ausführungsbeispielen können Erweiterungssymbole dafür genutzt werden, den Decoder für die Kernsymbole einzuschwingen um die Decodierbarkeit des Kernwortes zu erhöhen.

30

Fünftes Detailliertes Ausführungsbeispiel: Interleaving der Kernsequenz um Synchronisationssymbole

- Da die Kerninformation 122 von hoher Wichtigkeit ist, sollten die Symbole 136 (des  
35 Kernworts 130 = kanalcodierte Kerninformation 122) so zuverlässig wie möglich geschätzt und decodiert werden können. In einem Übertragungspaket befinden sich meist bekannte Symbole welche vom Empfänger zu Synchronisation genutzt werden, sog. Präambelsymbole

oder eine Synchronisationssequenz. Symbole, die direkt vor oder nach den Synchronisationssymbolen übertragen werden, können vom Decoder mit der höchsten Zuverlässigkeit decodiert werden, da Zeit oder Frequenzversätze zwischen Sender und Empfänger einen größeren Einfluss haben, je größer der zeitliche Abstand zwischen einem zu decodierenden Symbol und den bekannten Symbolen wird.

Um diesen Effekt für die Zuverlässigkeit des korrekten Empfangs der Kerninformation zu nutzen, ist es möglich, die Kernsymbole um die Synchronisationssequenzen herum anzuordnen.

Wird eine einzige Präambel für das ganze Telegramm verwendet, so können sich alle Kernsymbole am Anfang des Telegramms befinden, wie dies in Fig. 10 gezeigt ist.

Im Detail zeigt Fig. 10 in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den Subdatenpaketen den jeweiligen Kernsymbolen k0 bis k15 bzw. Erweiterungssymbolen e0 bis e39 Synchronisationssymbole 150 vorangestellt sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Die in Fig. 10 gezeigte Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3 entspricht im Wesentlichen der in Fig. 5 bis 8 erläuterten Aufteilung, wobei in den jeweiligen Subdatenpaketen (Kernsubdatenpaketen 140\_1 bis 140\_4 und Erweiterungsdatenpaketen 142\_1 bis 142\_3) den jeweiligen Kernsymbolen k0 bis k15 bzw. Erweiterungssymbolen e0 bis e39 Synchronisationssymbole 150 vorangestellt sind.

Mit anderen Worten, Fig. 10 zeigt Kernsymbole k0 bis k15 in den Kernsubpaketen 140\_1 bis 140\_3 am Anfang der Subpakete, da die Nähe zu Synchronisationssymbolen 150 die Schätzgenauigkeit für die Kernsymbole k0 bis k15 erhöht.

Für eine geteilte Übertragung der Subpakete, wie z.B. im Telegram-Splitting verfahren, ist eine Synchronisationssequenz in jedem Subpaket vorteilhaft. Hier werden die Kernsymbole in den Kernsubpaketen dann entsprechend in die Nähe der Prä-, Mid- oder Postamble

geschoben. Die Anordnung der Kernsymbole untereinander kann innerhalb des Subdatenpakets beliebig erfolgen. Von Subpaket zu Subpaket sollten die Kernsymbole wieder so angeordnet sein, dass der Abstand maximiert wird. Die Aufteilung in Symbole der Kernsubpakete bzw. Erweiterungssubpakete kann dann analog zu dem Vorgehen  
5 beschrieben in Ausführungsbeispiel zwei erfolgen.

Fig. 11 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in der Mitte der jeweiligen  
10 Subdatenpakete zwischen den jeweiligen Kernsymbolen k0 bis k15 bzw. Erweiterungssymbolen e0 bis e39 Synchronisationssymbole 150 vorhanden sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den  
15 jeweiligen Subdatenpaketen beschreibt.

Die in Fig. 11 gezeigte Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e39 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3 entspricht im Wesentlichen der in Fig. 5 bis 8 erläuterten Aufteilung, wobei in den  
20 jeweiligen Subdatenpaketen (Kernsubdatenpaketen 140\_1 bis 140\_4 und Erweiterungsdatenpaketen 142\_1 bis 142\_3) in der Mitte der jeweiligen Subdatenpakete Synchronisationssymbole 150 zwischen den jeweiligen Kernsymbolen k0 bis k15 bzw. Erweiterungssymbolen e0 bis e39 vorhanden sind.

25 Mit anderen Worten, Fig. 11 zeigt Kernsymbole k0 bis k15 in den Kernsubpaketen 140\_1 bis 140\_4 um die Midamble (Präambel, die in der Mitte eines Subdatenpaketes angeordnet ist) angeordnet, da die Nähe zu Synchronisationssymbolen 150 die Schätzgenauigkeit für die Kernsymbole k0 bis k15 erhöht.

30 Enthält ein Telegramm oder ein Subpaket mehrere getrennte Synchronisationssequenzen, so sollten die Kernsymbole wiederum um diese angeordnet werden.

Fig. 12 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e19 in Kernsubdatenpakete 140\_1 bis 140\_4 und  
35 Erweiterungssubdatenpakete 142\_1 bis 142\_5, wobei zwischen den Kernsubdatenpakete Synchronisationssubdatenpakete 152\_1 und 152\_2 mit Synchronisationssymbolen 150 angeordnet sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der jeweiligen

Subdatenpakete, während die Ordinate eine zeitliche Anordnung der Symbole in den jeweiligen Subdatenpaketen beschreibt.

Die in Fig. 12 gezeigte Aufteilung der Kernsymbole k0 bis k15 und Erweiterungssymbole e0 bis e19 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_35 entspricht im Wesentlichen der in Fig. 4 erläuterten Aufteilung.

Wie in Fig. 12 zu erkennen ist, kann ein erstes Synchronisationssubdatenpaket 152\_1 zwischen dem ersten und zweiten Kernsubdatenpaket 140\_1 und 140\_2 angeordnet sein, während ein zweites Synchronisationssubdatenpaket 152\_2 zwischen dem dritten und vierten Kernsubdatenpaket 140\_3 und 140\_4 angeordnet sein kann.

Mit anderen Worten, Fig. 12 zeigt Kernsymbole k0 bis k15 in Kernsubpaketen 140\_1 bis 140\_4 um zwei Synchronisationssubpakete 152\_1 und 152\_2 angeordnet, gefolgt von Erweiterungssubpaketen 142\_1 bis 142\_5.

Fig. 13 zeigt in einem Diagramm eine Aufteilung der Kernsymbole 136 und Erweiterungssymbole 138 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den jeweiligen Subdatenpaketen zwei voneinander beabstandete (Teil-)Synchronisationssymbolesequenzen vorhanden sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole + Synchronisationssymbole 150) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 13 zu erkennen ist, können in den Kernsubdatenpaketen 140\_1 bis 140\_4 die Kernsymbole 136 unmittelbar benachbart zu den Synchronisationssymbolen 150 angeordnet sein, während die Erweiterungssymbole 138 benachbart zu den Kernsymbolen 136 angeordnet sein können.

Mit anderen Worten, Fig. 13 zeigt Kernsymbole 136 gleichmäßig um zwei Synchronisationssequenzen 150 angeordnet.

Bei Ausführungsbeispielen können Symbole 136 der Kernwörter 130 so nahe wie möglich an Synchronisationssymbole 150 gesetzt werden.

Sechstes detailliertes Ausführungsbeispiel: Interleaving um verschieden zuverlässige getrennte Synchronisationssequenzen

Werden verschieden lange Synchronisationssequenzen verwendet, so ist die Zuverlässigkeit  
5 der Symbole unterschiedlich, je nach Güte der Synchronisationssequenz. Hierdurch können  
mehr Kernsymbole um die Synchronisationssequenz angeordnet mit höherer Güte gesetzt  
werden als um die Sequenz mit niedrigerer Güte, um die gleiche Fehlerwahrscheinlichkeit zu  
erhalten.

10 Fig. 14 zeigt in einem Diagramm eine Aufteilung der Kernsymbole 136 und  
Erweiterungssymbole 138 in Kernsubdatenpakete 140\_1 bis 140\_5 und  
Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den jeweiligen Subdatenpaketen  
zwei voneinander beabstandete unterschiedlich lange (Teil-  
)Synchronisationssymbolsequenzen vorhanden sind. Dabei beschreibt die Abszisse eine  
15 zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_5 +  
Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche  
Anordnung der Symbole (Kernsymbole + Erweiterungssymbole + Synchronisationssymbole  
150) in den jeweiligen Subdatenpaketen beschreibt.

20 Wie in Fig. 14 zu erkennen ist, können in den Kernsubdatenpaketen 140\_1 bis 140\_5 die  
Kernsymbole 136 unmittelbar benachbart zu den Synchronisationssymbolen 150 angeordnet  
sein, während die Erweiterungssymbole 138 benachbart zu den Kernsymbolen 136  
angeordnet sein können. Ferner können in den Kernsubdatenpaketen 140\_1 bis 140\_5 mehr  
Kernsymbole 136 unmittelbar benachbart zur längeren (Teil-  
25 )Synchronisationssymbolsequenz angeordnet sein, als zur kürzeren Teil-  
)Synchronisationssymbolsequenz.

Mit anderen Worten, Fig. 14 zeigt Kernsymbole 136 um zwei Synchronisationssequenzen  
150 angeordnet, die Anzahl der Kernsymbole 136 die eine Synchronisationssequenz 150  
30 umschließen kann Anhand der Güte der Synchronisationssequenz 150 (hier durch die Länge  
exemplarisch dargestellt) festgemacht werden.

Bei Ausführungsbeispielen können Symbole 136 der Kernwörter 130 so nahe wie möglich an  
Synchronisationssymbole 150 gesetzt werden (vgl. fünftes detailliertes Ausführungsbeispiel).

Bei Ausführungsbeispielen können Symbole 136 der Kernsequenz 130 nicht gleichmäßig um alle Synchronisationssymbolblöcke 150 verteilt werden, sondern entsprechend der zu erwartenden Sicherheit im Subpaket.

5 Siebtes detailliertes Ausführungsbeispiel: Interleaving um Synchronisation mit gemeinsamer Kanalcodierung von Kerninformation und Erweiterungsinformation

Werden Kerninformation und Erweiterungsinformation zusammen codiert (vgl. drittes detailliertes Ausführungsbeispiel), so ist es Vorteilhaft die Symbole 138 der Erweiterungsequenz 132, die zum einschwingen des Decoders verwendet werden, 10 möglichst nahe an die zuverlässigsten Positionen im Paket, also in die direkte Nähe der Präambeln zu schieben.

Bei Ausführungsbeispielen können Erweiterungssymbole 138 zum Einschwingen des Decoders (vgl. drittes detailliertes Ausführungsbeispiel) werden an die zuverlässigsten 15 Positionen im Paket gestellt (vgl. fünftes detailliertes Ausführungsbeispiel).

Achtes detailliertes Ausführungsbeispiel: Reencodierung der Kernsymbole und Nutzung als Synchronisationssymbole

20 Da die Kerninformation schon vor dem fertigen Empfang des ganzen Pakets verfügbar ist, kann die Information am Empfänger wieder kanalcodiert und gemappt werden. Durch dieses Vorgehen ist es dann möglich die Kernsymbole ebenfalls als Synchronisationssymbole anzusehen, da deren Werte jetzt bekannt sind. Mit diesem Wissen lässt sich die schätz- und 25 decodierbarkeit der Erweiterungssymbole in den Kernsubpaketen weiter verbessern.

Neuntes detailliertes Ausführungsbeispiel: Iterative Decodierung des Erweiterungswortes

Basierend auf dem achten detaillierten Ausführungsbeispiel ist es auch möglich, das 30 Erweiterungswort nach dem Gesamttempfang in kleinen Teilen zu decodieren und zu reencodieren um durch diese iterative Decodierung die Sicherheit bei der Schätzung der Erweiterungssymbole in den Subpaketen zu erhöhen.

Dieses Prinzip ist Beispielhaft in Fig. 15 bis 17 dargestellt.

35 Im Detail zeigt Fig. 15 in einem Diagramm eine Aufteilung der Kernsymbole 136 und Erweiterungssymbole 138 in Kernsubdatenpakete 140\_1 bis 140\_4 und

Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den jeweiligen Subdatenpaketen in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen 150 vorhanden sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate  
5 eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 15 zu erkennen ist, sind in den Kernsubdatenpaketen 140\_1 bis 140\_4 die Kernsymbole 136 und in den Erweiterungssubdatenpaketen 142\_1 bis 142\_3 ein Teil 138\*  
10 der Erweiterungssymbole 138 bedingt durch den geringen zeitlichen Abstand zu den Synchronisationssymbolen 150 gut zu decodieren.

Mit anderen Worten, Fig. 15 zeigt Kernsymbole 136 in Subpaket 0 bis 3 (140\_1 bis 140\_3) und Erweiterungssymbole 138\* in Subpaket 4 bis 6 (142\_1 bis 142\_3) sind auf Grund der  
15 Nähe zu den Synchronisationssymbolen 150 gut zu decodieren.

Fig. 16 zeigt in einem Diagramm eine Aufteilung der Kernsymbole 136 und Erweiterungssymbole 138 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den jeweiligen Subdatenpaketen in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen 150 vorhanden sind. Dabei  
20 beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 16 zu erkennen ist, können in den Kernsubdatenpaketen 140\_1 bis 140\_4 die Kernsymbole 136 nach der Decodierung reencodiert werden, um reencodierte Kernsymbole 136' zu erhalten, die zusammen mit den Synchronisationssymbolen 150 als bekannte  
25 Symbole angesehen werden können, die zur Decodierung der Erweiterungssymbole 138 genutzt werden können, so dass nun auch die Erweiterungssymbole auf Grund der Nähe zu den bekannten Symbolen (Synchronisationssymbolen 150 + reencodierte Kernsymbole 136')  
30 gut zu decodieren sind.

Mit anderen Worten, Fig. 15 zeigt dass nach reencodierung der Kernsymbole 136', diese als  
35 bekannt vorausgesetzt werden können und die Decodierbarkeit der benachbarten Erweiterungssymbole 138 in Subpaket 0 bis 3 (140\_1 bis 140\_3) erhöhen.

Fig. 17 zeigt in einem Diagramm eine Aufteilung der Kernsymbole 136 und Erweiterungssymbole 138 in Kernsubdatenpakete 140\_1 bis 140\_4 und Erweiterungssubdatenpakete 142\_1 bis 142\_3, wobei in den jeweiligen Subdatenpaketen in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen 150 vorhanden sind. Dabei  
5 beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

10 Wie in Fig. 17 zu erkennen ist, können in den Kernsubdatenpaketen 140\_1 bis 140\_4 der Teil 138\* der Erweiterungssymbole 138 nach der Decodierung reencodiert werden, um reencodierte Teil 138' der Erweiterungssymbole zu erhalten, die zusammen mit den Synchronisationssymbolen 150 als bekannte Symbole angesehen werden können, die zur Decodierung der restlichen Erweiterungssymbole 138 genutzt werden können, so dass nun  
15 auch die restlichen Erweiterungssymbole auf Grund der Nähe zu den bekannten Symbolen (Synchronisationssymbolen 150 + reencodierter Teil 138' der Erweiterungssymbole) gut zu decodieren sind.

Mit anderen Worten, Fig. 17 zeigt dass die zuverlässigen Erweiterungssymbole 138\* sind  
20 nach dem ersten iterativen decodierschritt auch reencodierbar und können als bekannt vorausgesetzt werden 138' und erweitern nun den zuverlässigen Bereich in den Subpaketen 4 bis 6 (142\_1 bis 142\_3).

Für den Interleaver bedeutet das, dass Symbole die bei der iterativen Decodierung früher  
25 gebraucht werden, näher an die Synchronisationssequenzen gestellt werden.

Bei Ausführungsbeispielen können Symbole des Erweiterungswortes näher an die Kernsymbole bzw. Synchronisationssymbole gestellt werden, je früher diese bei der iterativen Decodierung benötigt werden.  
30

#### Zehntes detailliertes Ausführungsbeispiel: Intra-Hop Bias vermeiden

Um bei der Anordnung der Kernsymbole 136 um die Synchronisationssymbole 150 ein eventuelles Ungleichgewicht zu vermeiden (z.B. durch Unsicherheiten in der Synchronisationssequenz, welche am Anfang jedoch nicht am Ende der Synchronisationssequenz auftreten) werden die Symbole in jedem zweiten Subpaket andersherum angeordnet. Ist in Subpaket 0 (140\_1) das Kernsymbol 0 (k<sub>0</sub>) über der  
35

Synchronisationssequenz (150) eingefügt worden, so ist das Kernsymbol 1 (k1) in Subpaket 1 (140\_2) unter der Synchronisationssequenz (150) anzuordnen.

Fig. 18 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 in Kernsubdatenpakete 140\_1 bis 140\_4, wobei in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_2 in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen 150 vorhanden sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 18 zu erkennen ist, können die Kernsymbole k0 bis k15 gleichmäßig verschachtelt auf die Kernsubdatenpakete 140\_1 bis 140\_4 aufgeteilt werden, so dass der Verlust eines der Kernsubdatenpakete 140\_1 bis 140\_3 nicht zum Totalverlust der Decodiermöglichkeit führt. Beispielsweise kann das erste Kernsubdatenpaket 140\_1 die Kernsymbole k0, k4, k8 und k12, das zweite Kernsubdatenpaket 140\_2 die Kernsymbole k1, k5, k9, k13, das dritte Kernsubdatenpaket 140\_3 die Kernsymbole k2, k6, k10 und k14, und das dritte Kernsubdatenpaket 140\_4 die Kernsymbole k3, k7, k11 und k15 enthalten, wobei die Kernsymbole k0 bis k3 in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_4 unmittelbar vor den Synchronisationssymbolen 150 angeordnet sind, und die Kernsymbole k4 bis k7 in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_4 unmittelbar nach den Synchronisationssymbolen 150 angeordnet sind.

Mit anderen Worten, Fig. 18 zeigt eine Aufteilung der Kernsymbole k0 bis k15 in Kernsubdatenpakete 140\_1 bis 140\_4 ohne Bias Vermeidung. Aufeinanderfolgende Symbole liegen auf der gleichen Seite (der Synchronisationssymbole 150).

Fig. 19 zeigt in einem Diagramm eine Aufteilung der Kernsymbole k0 bis k15 in Kernsubdatenpakete 140\_1 bis 140\_4, wobei in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_2 in der Mitte angeordnete (Teil-)Synchronisationssymbolsequenzen 150 vorhanden sind, wobei die Kernsymbole k0 bis k15 in aufeinander folgenden Kernsubdatenpaketen 140\_1 bis 140\_4 abwechselnd vor und nach den (Teil-)Synchronisationssymbolsequenzen 150 aufgeteilt sind. Dabei beschreibt die Abszisse eine zeitliche Anordnung der Subdatenpakete (Kernsubdatenpakete 140\_1 bis 140\_4 + Erweiterungssubdatenpakete 142\_1 bis 142\_3), während die Ordinate eine zeitliche Anordnung der Symbole (Kernsymbole + Erweiterungssymbole) in den jeweiligen Subdatenpaketen beschreibt.

Wie in Fig. 19 zu erkennen ist, können die Kernsymbole k0 bis k15 verschachtelt auf die Kernsubdatenpakete 140\_1 bis 140\_4 so aufgeteilt werden, dass die Kernsymbole k0 bis k15 in aufeinander folgenden Kernsubdatenpaketen 140\_1 bis 140\_4 abwechselnd vor und nach den (Teil-)Synchronisationssymbolsequenzen 150 angeordnet sind.

5

Beispielsweise kann das erste Kernsubdatenpaket 140\_1 die Kernsymbole k0, k4, k8 und k12, das zweite Kernsubdatenpaket 140\_2 die Kernsymbole k1, k5, k9, k13, das dritte Kernsubdatenpaket 140\_3 die Kernsymbole k2, k6, k10 und k14, und das vierte Kernsubdatenpaket 140\_4 die Kernsymbole k3, k7, k11 und k15 enthalten, wobei die Kernsymbole k0, k5, k2, und k7 in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_4 unmittelbar vor den Synchronisationssymbolen 150 angeordnet sind, und die Kernsymbole k4, k1, k6 und k3 in den jeweiligen Kernsubdatenpaketen 140\_1 bis 140\_4 unmittelbar nach den Synchronisationssymbolen 150 angeordnet sind.

10

15

Mit anderen Worten, Fig. 19 zeigt eine Aufteilung der Kernsymbole k0 bis k15 in Kernsubdatenpakete 140\_1 bis 140\_4 mit Bias Vermeidung. Aufeinanderfolgende Symbole werden abwechselnd um die Mitte (der Synchronisationssymbole 150) platziert.

#### Weitere Ausführungsbeispiele

20

Fig. 20 zeigt ein Flussdiagramm eines Verfahrens 200 zum Senden von Kerndaten und Erweiterungsdaten. Das Verfahren 200 umfasst einen Schritt 202 des Codierens von Kerndaten, um codierte Kerndaten zu erhalten. Ferner umfasst das Verfahren 200 einen Schritt 204 des Verschachtelns und Aufteilens der codierten Kerndaten auf eine Mehrzahl von Kernsubdatenpakete. Ferner umfasst das Verfahren 200 einen Schritt 206 des Codierens von Erweiterungsdaten, um codierte Erweiterungsdaten zu erhalten. Ferner umfasst das Verfahren 200 einen Schritt 208 des Verschachtelns und Aufteilens der codierten Erweiterungsdaten auf eine Mehrzahl von Erweiterungsdatenpakete. Ferner umfasst das Verfahren 200 einen Schritt 210 des Sendens der Kernsubdatenpakete und Erweiterungsdatenpakete.

25

30

35

Fig. 21 zeigt ein Flussdiagramm eines Verfahrens 220 zum Empfangen von Kerndaten und Erweiterungsdaten. Das Verfahren 220 umfasst einen Schritt 222 des Empfangens von Kernsubdatenpaketen und Erweiterungssubdatenpaketen, wobei die Kernsubdatenpakete Kerndaten enthalten, die über die Kernsubdatenpakete verschachtelt verteilt sind, und wobei die Erweiterungssubdatenpakete Erweiterungsdaten enthalten, die über die Erweiterungssubdatenpakete verschachtelt verteilt sind. Ferner umfasst das Verfahren 220

einen Schritt 224 des Decodierens zumindest eines Teils der codierten Kerndaten, um eine Information bezüglich der Erweiterungsdatenpakete zu erhalten, wobei die Erweiterungsdatenpakete unter Verwendung der Information empfangen werden.

5 Obwohl Ausführungsbeispiele beschrieben wurden in denen die Erweiterungssubpakete zeitlich nach den Kernsubpaketen angeordnet sind, sei darauf hingewiesen, dass die vorliegende Erfindung nicht auf solche Ausführungsbeispiele beschränkt ist. Vielmehr können die Erweiterungssubpakete auch zeitlich vor oder zeitgleich zu den Kernsubpaketen angeordnet sein, z.B. wenn ein Empfangspuffer genutzt wird.

10

Ausführungsbeispiele verschachteln die Daten in Subpaketblöcke. Ausführungsbeispiele befassen sich mit einer Methode, Information so über Subpakete zu verteilen, dass der Verlust eines oder mehrerer Subpaketes die Decodierbarkeit des Gesamtpaketes so wenig wie möglich beeinflusst.

15

Ausführungsbeispiele nutzen eine Kerninformation mit Längenangabe. Zudem kann durch das geschickte Verteilen ein Teil der Information schon vor Empfang des Gesamtpaketes (Kernsubdatenpakete + Erweiterungssubdatenpakete) decodiert und verarbeitet werden. Dies erlaubt es auch, z.B. eine variable Anzahl an Subpaketen zu verschicken, wenn dem Empfänger über diese Information die Anzahl der Subpakete bzw. die Länge des zu empfangenden Pakets mitgeteilt wird.

20

Bei Ausführungsbeispielen ist die Kern-/ bzw. Erweiterungsinformation auch mit kurzer Länge gut schützbar. Bisherige Systeme, die einen Core-Block und eine optionale Erweiterungsequenz haben, haben Probleme, die Erweiterungsequenz oder die Kernsequenz mit einem guten Fehlerschutz zu schützen, falls diese nur sehr kurz sind. Es ist bekannt [S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," TMO progress report, vol. 42, p. 133, 1998.], dass sich kleine Datenmengen nur schlecht schützen lassen. Ein gemeinsames Kanalcodieren beider Teile erhöht die

25

30

Gesamtlänge der Daten und somit den Fehlerschutz.

35

Ausführungsbeispiele erhöhen eine Entscheidungssicherheit der Kerninformationssymbole. Bisherige Systeme machen sich nicht zu nutzen, dass die Entscheidungssicherheit der Symbole in der Nähe von bekannten Symbolen verbessert wird. Diese Erfindung beschreibt eine Methode die Verschachtelung der Symbole derart vorzunehmen, dass die Kernsymbole eine höhere Entscheidungssicherheit beim Empfang erhalten indem deren

Position innerhalb der Subpakete so gewählt wird, dass diese sich zuverlässiger decodieren lassen.

5 Ausführungsbeispiele beziehen sich auf eine Methode zum intelligenten Verschachteln der Symbole in einer Übertragung. Die Verschachtelung basiert auf Subpaketen über welche die Symbole verteilt werden, was den Verlust einzelner Subpakete bei der Übertragung erlaubt. Des Weiteren können in einem Subpaket die Symbole so angeordnet werden, dass eine Kerninformation vor dem abgeschlossenen Empfang aller Subpakete extrahiert werden kann. Ausführungsbeispiele beschreiben weiter wie durch die Anordnung der Symbole die  
10 Zuverlässigkeit der Übertragung der Kerninformation erhöht werden kann.

Ausführungsbeispiele schaffen ein System zur Übertragung von Daten zwischen verschiedenen Teilnehmern eines Funknetzwerkes, bei der die zu übertragenden Daten in eine Kerninformation und eine Erweiterungsinformation aufgeteilt werden.  
15

Ausführungsbeispiele können auf traditionelle Übertragungsverfahren und auf Telegram-Splitting basierte Übertragungsverfahren, also der Aufteilung der Übertragung in mehrere zeitlich versetzt gesendete Subdatenpakete, angewendet werden.

20 Ausführungsbeispiele ermöglichen eine Decodierung (z.B. durch einen Empfänger) eines Teils der zu übertragenden Information (z.B. der Kerninformation) schon vor dem Empfang des Gesamtdatenpakets (also der Erweiterungsinformation).

Ausführungsbeispiele ermöglichen ein verbessertes (oder sogar optimales) Verschachteln der Daten.  
25

Bei Ausführungsbeispielen ist die Kerninformation vor abgeschlossenem Empfang verfügbar.

Bei Ausführungsbeispielen ist die Kerninformation auch mit kurzer Länge gut schützbar.  
30

Ausführungsbeispiele ermöglichen eine Erhöhung der Entscheidungssicherheit der Kerninformationssymbole.

35 Obwohl manche Aspekte im Zusammenhang mit einer Vorrichtung beschrieben wurden, versteht es sich, dass diese Aspekte auch eine Beschreibung des entsprechenden Verfahrens darstellen, sodass ein Block oder ein Bauelement einer Vorrichtung auch als ein entsprechender Verfahrensschritt oder als ein Merkmal eines Verfahrensschrittes zu

verstehen ist. Analog dazu stellen Aspekte, die im Zusammenhang mit einem oder als ein Verfahrensschritt beschrieben wurden, auch eine Beschreibung eines entsprechenden Blocks oder Details oder Merkmals einer entsprechenden Vorrichtung dar. Einige oder alle der Verfahrensschritte können durch einen Hardware-Apparat (oder unter Verwendung eines Hardware-Apparats), wie zum Beispiel einen Mikroprozessor, einen programmierbaren Computer oder eine elektronische Schaltung ausgeführt werden. Bei einigen Ausführungsbeispielen können einige oder mehrere der wichtigsten Verfahrensschritte durch einen solchen Apparat ausgeführt werden.

10 Je nach bestimmten Implementierungsanforderungen können Ausführungsbeispiele der Erfindung in Hardware oder in Software implementiert sein. Die Implementierung kann unter Verwendung eines digitalen Speichermediums, beispielsweise einer Floppy-Disk, einer DVD, einer Blu-ray Disc, einer CD, eines ROM, eines PROM, eines EPROM, eines EEPROM oder eines FLASH-Speichers, einer Festplatte oder eines anderen magnetischen oder optischen Speichers durchgeführt werden, auf dem elektronisch lesbare Steuersignale gespeichert sind, die mit einem programmierbaren Computersystem derart zusammenwirken können oder zusammenwirken, dass das jeweilige Verfahren durchgeführt wird. Deshalb kann das digitale Speichermedium computerlesbar sein.

20 Manche Ausführungsbeispiele gemäß der Erfindung umfassen also einen Datenträger, der elektronisch lesbare Steuersignale aufweist, die in der Lage sind, mit einem programmierbaren Computersystem derart zusammenzuwirken, dass eines der hierin beschriebenen Verfahren durchgeführt wird.

25 Allgemein können Ausführungsbeispiele der vorliegenden Erfindung als Computerprogrammprodukt mit einem Programmcode implementiert sein, wobei der Programmcode dahingehend wirksam ist, eines der Verfahren durchzuführen, wenn das Computerprogrammprodukt auf einem Computer abläuft.

30 Der Programmcode kann beispielsweise auch auf einem maschinenlesbaren Träger gespeichert sein.

Andere Ausführungsbeispiele umfassen das Computerprogramm zum Durchführen eines der hierin beschriebenen Verfahren, wobei das Computerprogramm auf einem maschinenlesbaren Träger gespeichert ist.

Mit anderen Worten ist ein Ausführungsbeispiel des erfindungsgemäßen Verfahrens somit ein Computerprogramm, das einen Programmcode zum Durchführen eines der hierin beschriebenen Verfahren aufweist, wenn das Computerprogramm auf einem Computer abläuft.

5

Ein weiteres Ausführungsbeispiel der erfindungsgemäßen Verfahren ist somit ein Datenträger (oder ein digitales Speichermedium oder ein computerlesbares Medium), auf dem das Computerprogramm zum Durchführen eines der hierin beschriebenen Verfahren aufgezeichnet ist. Der Datenträger, das digitale Speichermedium oder das computerlesbare  
10 Medium sind typischerweise gegenständlich und/oder nicht-vergänglich bzw. nicht-vorübergehend.

15

Ein weiteres Ausführungsbeispiel des erfindungsgemäßen Verfahrens ist somit ein Datenstrom oder eine Sequenz von Signalen, der bzw. die das Computerprogramm zum Durchführen eines der hierin beschriebenen Verfahren darstellt bzw. darstellen. Der Datenstrom oder die Sequenz von Signalen kann bzw. können beispielsweise dahingehend konfiguriert sein, über eine Datenkommunikationsverbindung, beispielsweise über das Internet, transferiert zu werden.

20

Ein weiteres Ausführungsbeispiel umfasst eine Verarbeitungseinrichtung, beispielsweise einen Computer oder ein programmierbares Logikbauelement, die dahingehend konfiguriert oder angepasst ist, eines der hierin beschriebenen Verfahren durchzuführen.

25

Ein weiteres Ausführungsbeispiel umfasst einen Computer, auf dem das Computerprogramm zum Durchführen eines der hierin beschriebenen Verfahren installiert ist.

30

Ein weiteres Ausführungsbeispiel gemäß der Erfindung umfasst eine Vorrichtung oder ein System, die bzw. das ausgelegt ist, um ein Computerprogramm zur Durchführung zumindest eines der hierin beschriebenen Verfahren zu einem Empfänger zu übertragen. Die Übertragung kann beispielsweise elektronisch oder optisch erfolgen. Der Empfänger kann beispielsweise ein Computer, ein Mobilgerät, ein Speichergerät oder eine ähnliche Vorrichtung sein. Die Vorrichtung oder das System kann beispielsweise einen Datei-Server zur Übertragung des Computerprogramms zu dem Empfänger umfassen.

35

Bei manchen Ausführungsbeispielen kann ein programmierbares Logikbauelement (beispielsweise ein feldprogrammierbares Gatterarray, ein FPGA) dazu verwendet werden, manche oder alle Funktionalitäten der hierin beschriebenen Verfahren durchzuführen. Bei

manchen Ausführungsbeispielen kann ein feldprogrammierbares Gatterarray mit einem Mikroprozessor zusammenwirken, um eines der hierin beschriebenen Verfahren durchzuführen. Allgemein werden die Verfahren bei einigen Ausführungsbeispielen seitens einer beliebigen Hardwarevorrichtung durchgeführt. Diese kann eine universell einsetzbare Hardware wie ein Computerprozessor (CPU) oder eine Grafikkarte (GPU) sein oder für das Verfahren spezifische Hardware, wie beispielsweise ein ASIC.

Die hierin beschriebenen Vorrichtungen können beispielsweise unter Verwendung eines Hardware-Apparats, oder unter Verwendung eines Computers, oder unter Verwendung einer Kombination eines Hardware-Apparats und eines Computers implementiert werden.

Die hierin beschriebenen Vorrichtungen, oder jedwede Komponenten der hierin beschriebenen Vorrichtungen können zumindest teilweise in Hardware und/oder in Software (Computerprogramm) implementiert sein.

Die hierin beschriebenen Verfahren können beispielsweise unter Verwendung eines Hardware-Apparats, oder unter Verwendung eines Computers, oder unter Verwendung einer Kombination eines Hardware-Apparats und eines Computers implementiert werden.

Die hierin beschriebenen Verfahren, oder jedwede Komponenten der hierin beschriebenen Verfahren können zumindest teilweise durch Hardware und/oder durch Software ausgeführt werden.

Die oben beschriebenen Ausführungsbeispiele stellen lediglich eine Veranschaulichung der Prinzipien der vorliegenden Erfindung dar. Es versteht sich, dass Modifikationen und Variationen der hierin beschriebenen Anordnungen und Einzelheiten anderen Fachleuten einleuchten werden. Deshalb ist beabsichtigt, dass die Erfindung lediglich durch den Schutzzumfang der nachstehenden Patentansprüche und nicht durch die spezifischen Einzelheiten, die anhand der Beschreibung und der Erläuterung der Ausführungsbeispiele hierin präsentiert wurden, beschränkt sei.

Patentansprüche

1. 5 Datensender (100), der ausgebildet ist, um Kerndaten (122) zu codieren und verschachteln und auf eine Mehrzahl von Kernsubdatenpakete aufzuteilen, wobei der Datensender (100) ausgebildet ist, um Erweiterungsdaten (124) zu codieren und verschachtelt auf eine Mehrzahl von Erweiterungssubdatenpakete aufzuteilen, wobei  
10 zumindest ein Teil der in den Kernsubdatenpaketen enthaltenen codierten Kerndaten (130) für einen Empfang der codierten Erweiterungsdaten (132) oder Erweiterungssubdatenpakete (142\_1:142\_m) erforderlich ist.
2. 15 Datensender (100) nach Anspruch 1, wobei der Datensender (100) ausgebildet ist, um die codierten Kerndaten (130) auf die Kernsubdatenpakete (140\_1:140\_n) derart aufzuteilen, dass auch bei Übertragungsverlust eines oder mehrerer der Kernsubdatenpakete (140\_1:140\_n) eine empfängerseitige Decodierung der codierten Kerndaten (130) basierend auf den anderen Kernsubdatenpaketen (140\_1:140\_n) möglich ist.
3. 20 Datensender (100) nach einem der Ansprüche 1 bis 2, wobei der Datensender (100) ausgebildet ist, um die codierten Kerndaten (130) auf die Kernsubdatenpakete (140\_1:140\_n) derart aufzuteilen, dass ein Abstand der codierten Kerndaten (130) in Bezug auf eine Einfluslänge eines für die Codierung der Kerndaten (122) verwendeten Codes erhöht ist.
- 25 4. 30 Datensender (100) nach einem der Ansprüche 1 bis 3, wobei der Datensender (100) ausgebildet ist, um, falls eine Länge der codierten Kerndaten (130) zur Auffüllung der Kernsubdatenpakete (140\_1:140\_n) nicht ausreicht, die codierten Erweiterungsdaten verschachtelt auf die Kernsubdatenpakete (140\_1:140\_n) aufzuteilen, um die Kernsubdatenpakete (140\_1:140\_n) aufzufüllen.
- 35 5. 35 Datensender (100) nach Anspruch 4, wobei der Datensender (100) ausgebildet ist, um bei der Auffüllung der Kernsubdatenpakete (140\_1:140\_n) die codierten Erweiterungsdaten (132) derart auf die Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) aufzuteilen, dass ein Abstand der codierten Erweiterungsdaten (132) in Bezug auf eine Einfluslänge eines für die Codierung der Erweiterungsdaten (124) verwendeten Codes erhöht ist.

6. Datensender (100) nach einem der Ansprüche 4 bis 5, wobei der Datensender (100) ausgebildet ist, um bei der Auffüllung der Kernsubdatenpakete (140\_1:140\_n) die codierten Erweiterungsdaten (132) derart auf die Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) aufzuteilen, dass die  
5 Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) gleichmäßig aufgefüllt werden.
7. Datensender (100) nach einem der Ansprüche 4 bis 5, wobei der Datensender (100) ausgebildet ist, um bei der Auffüllung der Kernsubdatenpakete (140\_1:140\_n) die  
10 codierten Erweiterungsdaten (124) derart auf die Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) aufzuteilen, dass die Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) ungleichmäßig aufgefüllt werden.
8. Datensender (100) nach einem der Ansprüche 1 bis 7, wobei der Datensender (100) ausgebildet ist, um die codierten Kerndaten (130) auf eine feste oder vorgegebene Anzahl von Kernsubdatenpakete (140\_1:140\_n) aufzuteilen.
9. Datensender (100) nach einem der Ansprüche 1 bis 8, wobei der Datensender (100) ausgebildet ist, um eine Anzahl von Erweiterungssubdatenpaketen (142\_1:142\_m) in  
20 Abhängigkeit von einer Länge der Erweiterungsdaten anzupassen.
10. Datensender (100) nach einem der Ansprüche 1 bis 9, wobei der Datensender (100) ausgebildet ist, um die Kerndaten (122) und die Erweiterungsdaten (124) gemeinsam  
25 zu codieren.
11. Datensender (100) nach Anspruch 10, wobei die Kerndaten (122) und die Erweiterungsdaten (124) gemeinsam so codiert werden, dass eine Decodierung der codierten Kerndaten (130) zumindest einen Teil der Kerndaten liefert.  
30
12. Datensender (100) nach einem der Ansprüche 1 bis 9, wobei der Datensender (100) ausgebildet ist, um die Kerndaten (122) und die Erweiterungsdaten (124) unabhängig voneinander zu codieren.
13. Datensender (100) nach einem der Ansprüche 1 bis 12, wobei der Datensender (100) ausgebildet ist, um die uncodierten Kerndaten (122) mit Erweiterungsdaten (124) zu  
35

füllen, so dass die Erweiterungsdaten zeitlich vor den Kerndaten angeordnet sind und die Sicherheit bei der Decodierung der Kerndaten erhöht ist.

- 5 14. Datensender (100) nach einem der Ansprüche 1 bis 13, wobei der Datensender ausgebildet ist, um die codierten Erweiterungsdaten (132) ferner auf zumindest einen Anteil der Kernsubdatenpakete (140\_1:140\_n) aufzuteilen, so dass in den jeweiligen Kernsubdatenpaketen (140\_1:140\_n) ein Teil der codierten Erweiterungsdaten zeitlich vor den codierten Kerndaten (130) angeordnet sind, so dass eine Sicherheit bei der Decodierung der codierten Kerndaten (130) erhöht ist.
- 10 15. Datensender (100) nach einem der Ansprüche 1 bis 14, wobei der Datensender (100) ausgebildet ist, um zumindest einen Anteil der Kernsubdatenpakete (140\_1:140\_n) mit Synchronisationsdaten (150) zu versehen.
- 15 16. Datensender (100) nach Anspruch 15, wobei der Datensender (100) ausgebildet ist, um die in den jeweiligen Kernsubdatenpaketen (140\_1:140\_n) die codierten Kerndaten (130) zeitlich benachbart zu den Synchronisationsdaten (150) anzuordnen.
- 20 17. Datensender (100) nach Anspruch 15, wobei der Datensender (100) ausgebildet ist, um in zeitlich aufeinanderfolgenden Kernsubdatenpaketen (140\_1:140\_n) die Kerndaten (130) abwechselnd vor und nach den Synchronisationsdaten (150) anzuordnen.
- 25 18. Datensender (100) nach einem der Ansprüche 15 bis 17, wobei der Datensender (100) ausgebildet ist, um in den jeweiligen Kernsubdatenpaketen (140\_1:140\_n) die Synchronisationsdaten (150) zeitlich so anzuordnen, dass diese unmittelbar benachbart zu den codierten Erweiterungsdaten (132) und unmittelbar benachbart zu den codierten Kerndaten (130) angeordnet sind.
- 30 19. Datensender (100) nach einem der Ansprüche 1 bis 18, wobei der Datensender (100) ferner ausgebildet ist, um reine Synchronisationssubdatenpakete (152\_1,152\_2) auszusenden.
- 35 20. Datensender (100) nach Anspruch 19, wobei der Datensender (100) ausgebildet ist, die Kernsubdatenpakete (140\_1:140\_n) und die Synchronisationssubdatenpakete (152\_1,152\_2) so auszusenden, dass die Kernsubdatenpakete (140\_1:140\_n) und

die Synchronisationssubdatenpakete (152\_1,152\_2) zeitlich benachbart zueinander angeordnet sind.

21. 5 Datenempfänger (110), der ausgebildet ist, um Kernsubdatenpakete (140\_1:140\_n) und Erweiterungssubdatenpakete (142\_1:142\_m) zu empfangen, wobei die Kernsubdatenpakete (140\_1:140\_n) codierte Kerndaten (130) enthalten, die über die Kernsubdatenpakete (140\_1:140\_n) verschachtelt verteilt sind, und wobei die Erweiterungssubdatenpakete (142\_1:142\_m) codierte Erweiterungsdaten (132) enthalten, die über die Erweiterungssubdatenpakete (142\_1:142\_m) verschachtelt verteilt sind; 10

wobei der Datenempfänger (110) ausgebildet ist, um zumindest einen Teil der codierten Kerndaten (130) zu decodieren, um eine Information bezüglich der codierten Erweiterungsdaten (132) oder Erweiterungssubdatenpakete (142\_1:142\_m) zu erhalten; 15

wobei der Datenempfänger (110) ausgebildet ist, um die Erweiterungsdatenpakete (142\_1:142\_m) unter Verwendung der Information zu empfangen.

20 22. Datenempfänger (110) nach Anspruch 21, wobei dem Datenempfänger (110) eine Anzahl der Kernsubdatenpakete (140\_1:140\_n) bekannt ist.

23. 25 Datenempfänger (110) nach einem der Ansprüche 21 bis 22, wobei die Information bezüglich der Erweiterungssubdatenpakete eine Anzahl der Erweiterungssubdatenpakete (142\_1:142\_m) umfasst.

24. 30 Datenempfänger (110) nach einem der Ansprüche 21 bis 23, wobei die codierten Kerndaten (130) auf die Kernsubdatenpakete (140\_1:140\_n) derart aufgeteilt sind, dass auch bei Übertragungsverlust eines oder mehrerer der Kernsubdatenpakete (140\_1:140\_n) eine empfängerseitige Decodierung der codierten Kerndaten (130) basierend auf den anderen Kernsubdatenpaketen (140\_1:140\_n) möglich ist;

wobei der Datenempfänger (110) ausgebildet ist, um zumindest einen Anteil der Kernsubdatenpakete (140\_1:140\_n) zu empfangen und zu decodieren, um die Kerndaten (122) zu erhalten. 35

25. Datenempfänger (110) nach einem der Ansprüche 21 bis 24, wobei zumindest ein Anteil der Kernsubdatenpakete (140\_1:140\_n) mit Synchronisationsdaten (150) versehen ist;

5 wobei der Datenempfänger (110) ausgebildet ist, um die Kernsubdatenpakete (140\_1:140\_n) basierend auf zumindest einem Anteil der Synchronisationsdaten (150) in einem Empfangsdatenstrom zu detektieren.

10 26. Datenempfänger (110) nach einem der Ansprüche 21 bis 25, wobei der Datenempfänger (110) ausgebildet ist, um reine Synchronisationssubdatenpakete (152\_1,152\_2) zu empfangen, und um die Kernsubdatenpakete (140\_1:140\_n) basierend auf zumindest einem Anteil der Synchronisationssubdatenpakete (152\_1,152\_2) in einem Empfangsdatenstrom zu detektieren.

15 27. Datenempfänger (110) nach einem der Ansprüche 21 bis 26, wobei der Datenempfänger (110) ausgebildet ist, um zumindest einen Teil der decodierten Kerndaten zu reencodieren, um reencodierte Kerndaten zu erhalten;

20 wobei der Datenempfänger (110) ausgebildet ist, um zumindest einen Teil der codierten Erweiterungsdaten unter Verwendung der reencodierten Kerndaten zu decodieren.

25 28. Datenempfänger (110) nach einem der Ansprüche 21 bis 27, wobei der Datenempfänger (110) ausgebildet ist, um einen ersten Teil der codierten Erweiterungsdaten zu decodieren und zu reencodieren, um einen ersten Teil von reencodierten Erweiterungsdaten zu erhalten;

30 wobei der Datenempfänger (110) ausgebildet ist, um einen zweiten Teil der codierten Erweiterungsdaten unter Verwendung des ersten Teils von reencodierten Erweiterungsdaten zu decodieren.

29. System, mit folgenden Merkmalen:

35 einem Datensender (100) nach einem der Ansprüche 1 bis 20; und

einem Datenempfänger (110) nach einem der Ansprüche 21 bis 28.

30. Verfahren (200), mit folgenden Schritten:

Codieren (202) von Kerndaten, um codierte Kerndaten zu erhalten;

5 Verschachteln (204) und Aufteilen der codierten Kerndaten auf eine Mehrzahl von Kernsubdatenpakete;

Codieren (206) von Erweiterungsdaten, um codierte Erweiterungsdaten zu erhalten;

10 Verschachteln (208) und Aufteilen der codierten Erweiterungsdaten auf eine Mehrzahl von Erweiterungsdatenpakete;

Senden (210) der Kernsubdatenpakete und Erweiterungsdatenpakete.

15 31. Verfahren (220), mit folgenden Schritten:

Empfangen (222) von Kernsubdatenpaketen und Erweiterungssubdatenpaketen, wobei die Kernsubdatenpakete Kerndaten enthalten, die über die Kernsubdatenpakete verschachtelt verteilt sind, und wobei die Erweiterungssubdatenpakete Erweiterungsdaten enthalten, die über die Erweiterungssubdatenpakete verschachtelt verteilt sind;

20 Decodieren (224) zumindest eines Teils der codierten Kerndaten, um eine Information bezüglich der Erweiterungsdatenpakete zu erhalten;

25 wobei die Erweiterungsdatenpakete unter Verwendung der Information empfangen werden.

30 32. Computerprogramm zur Durchführung des Verfahrens nach Anspruch 30 oder 31.

35 33. Datensender (100), der ausgebildet ist, um Daten fester Länge zu codieren und verschachtelt auf zumindest eine Untermenge einer Mehrzahl von Subdatenpakete aufzuteilen, wobei der Datensender (100) ferner ausgebildet ist, um Daten variabler Länge zu codieren und verschachtelt auf zumindest eine Untermenge der Mehrzahl von Subdatenpakete aufzuteilen, so dass zumindest eins der Mehrzahl von Subdatenpaketen einen Teil der Daten fester Länge und einen Teil der Daten variabler Länge aufweist.

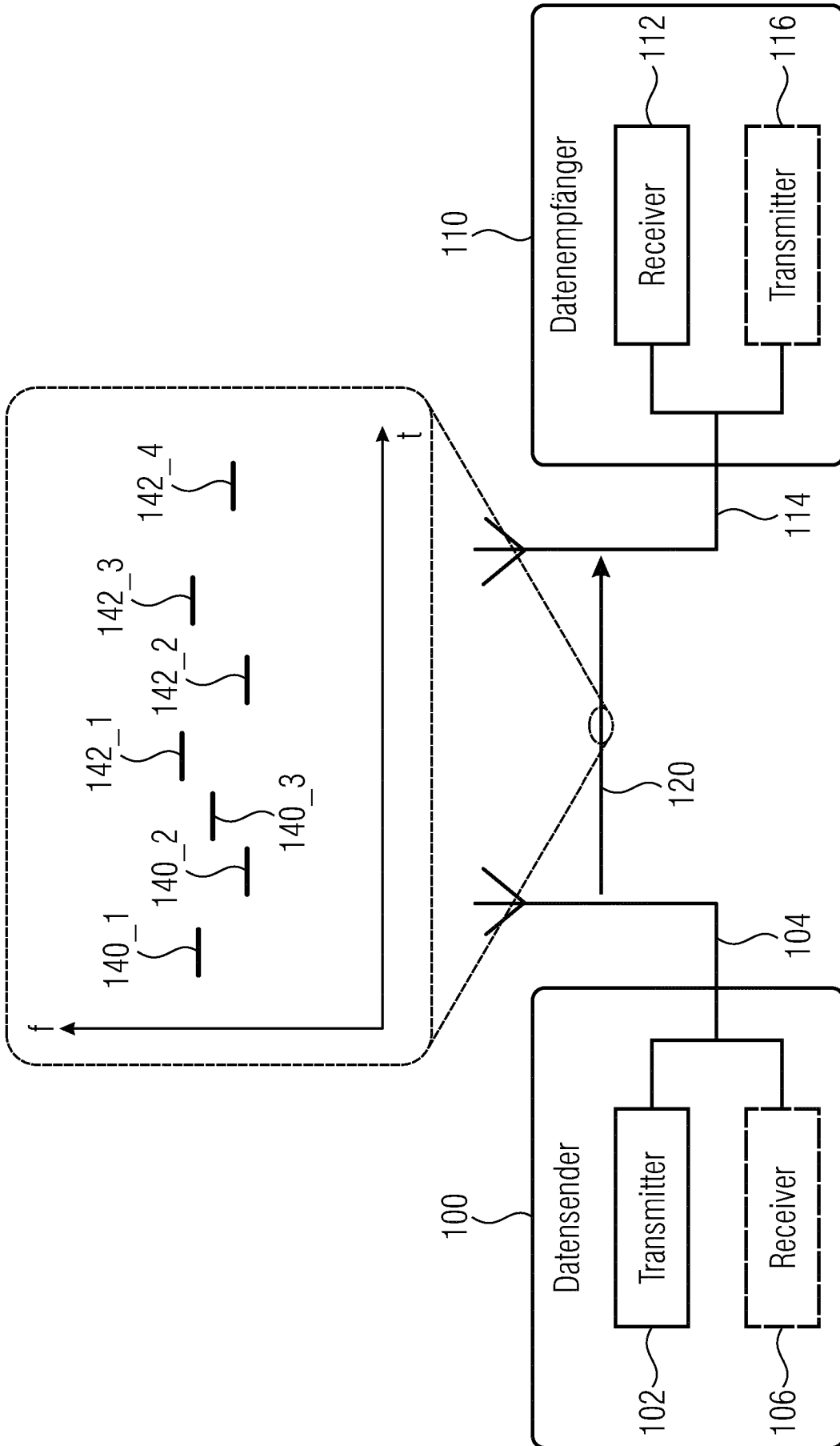


Fig. 1

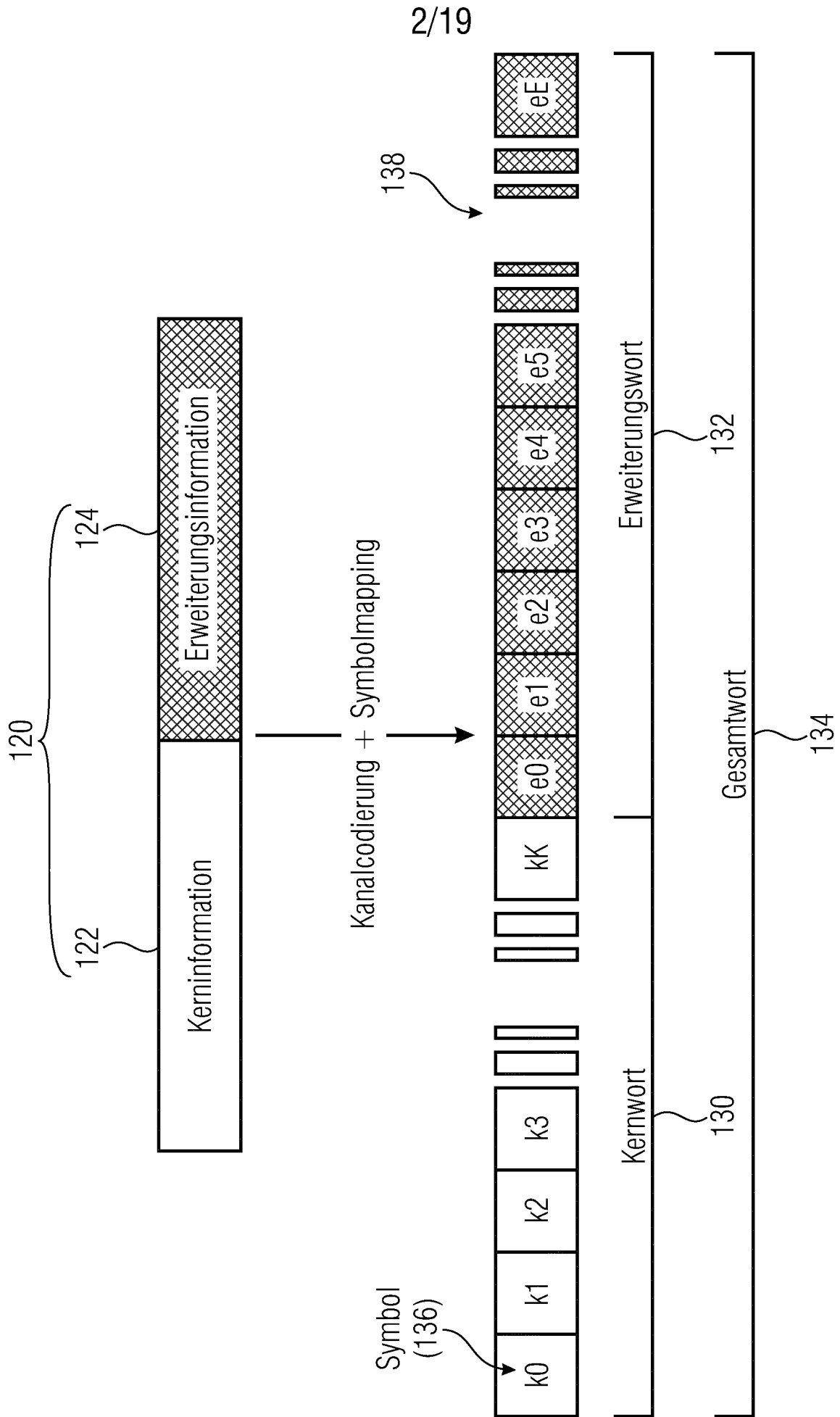


Fig. 2

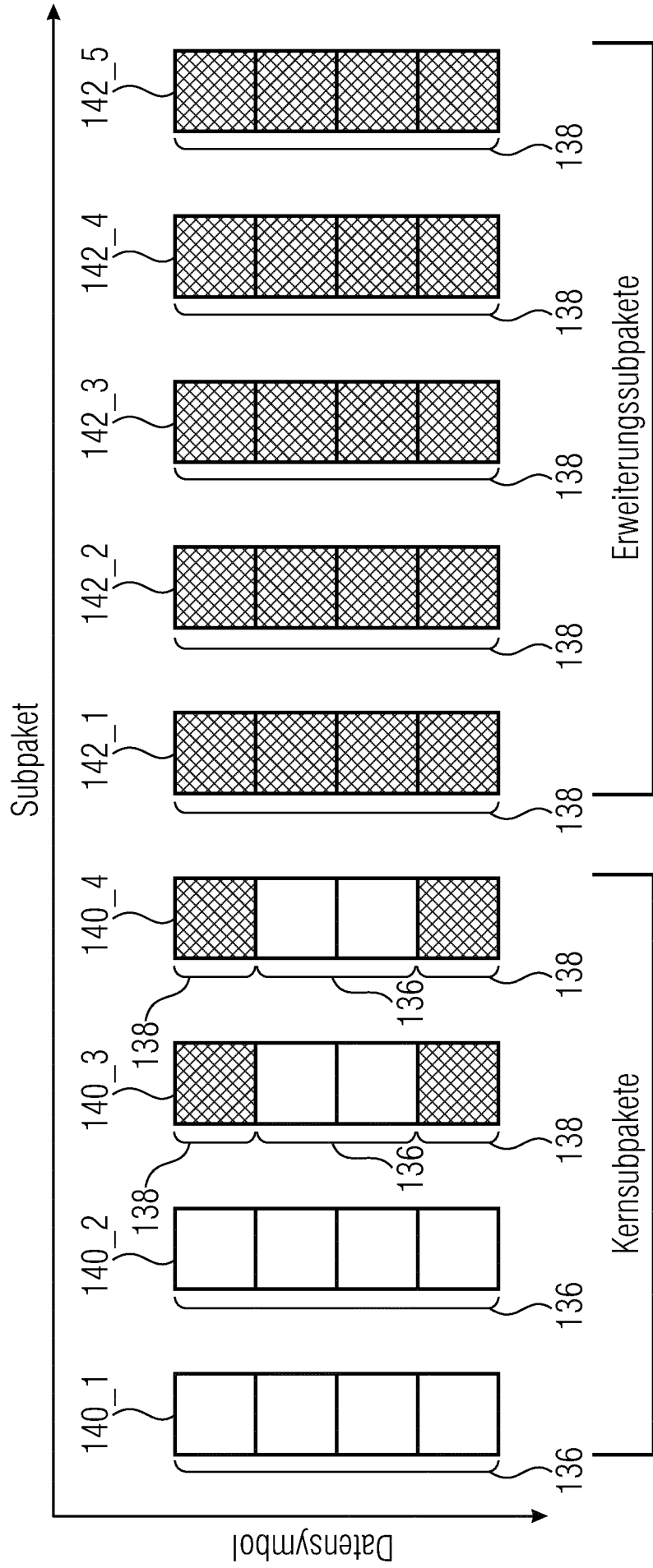


Fig. 3

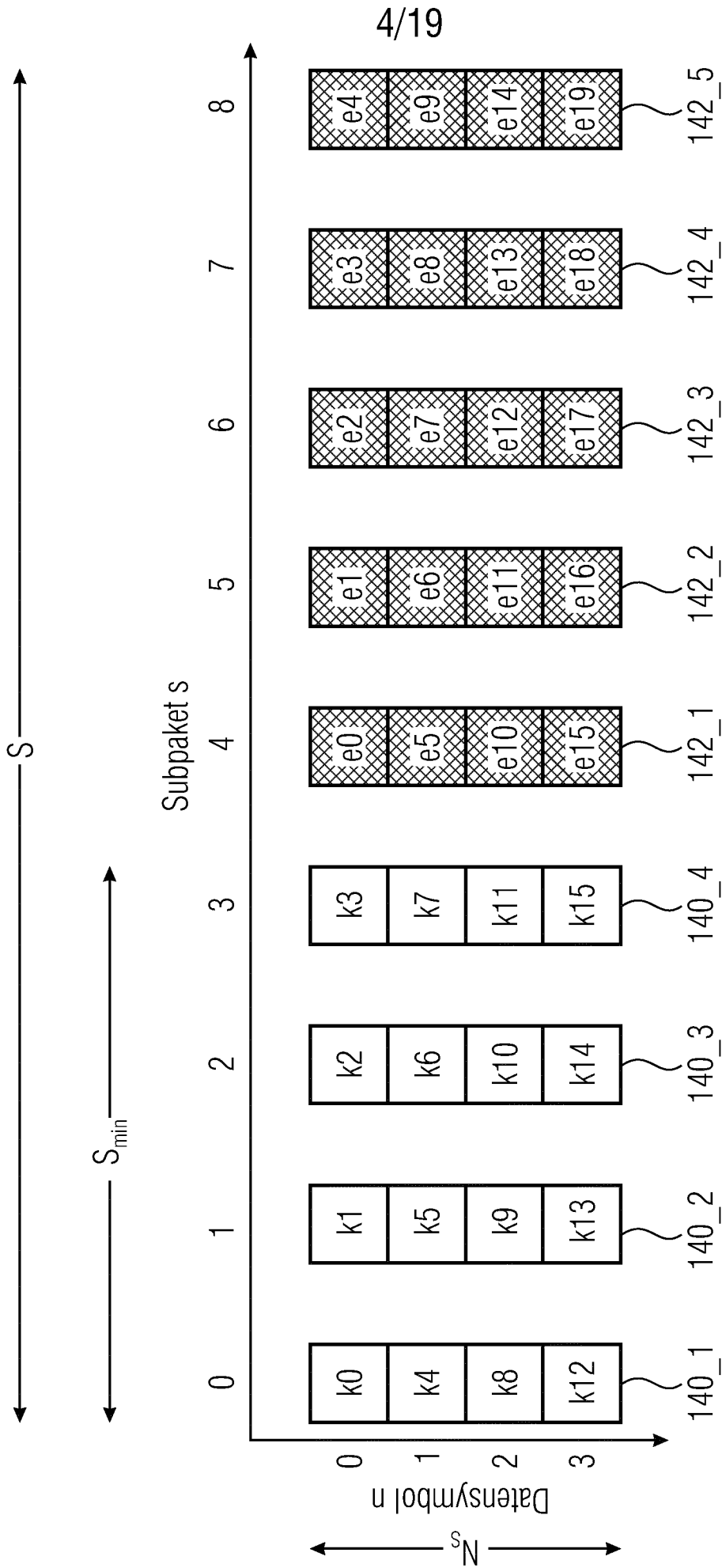


Fig. 4

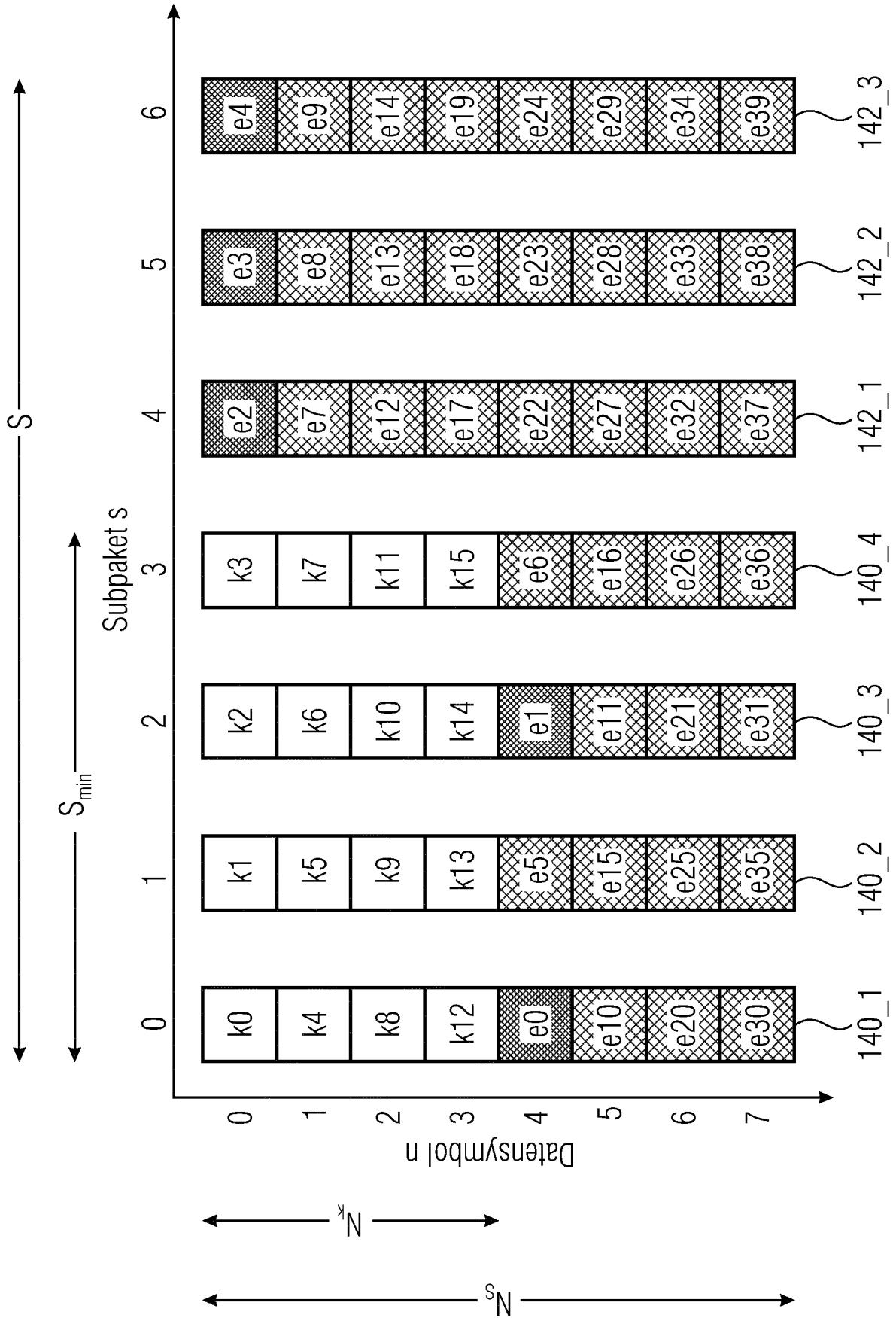


Fig. 5

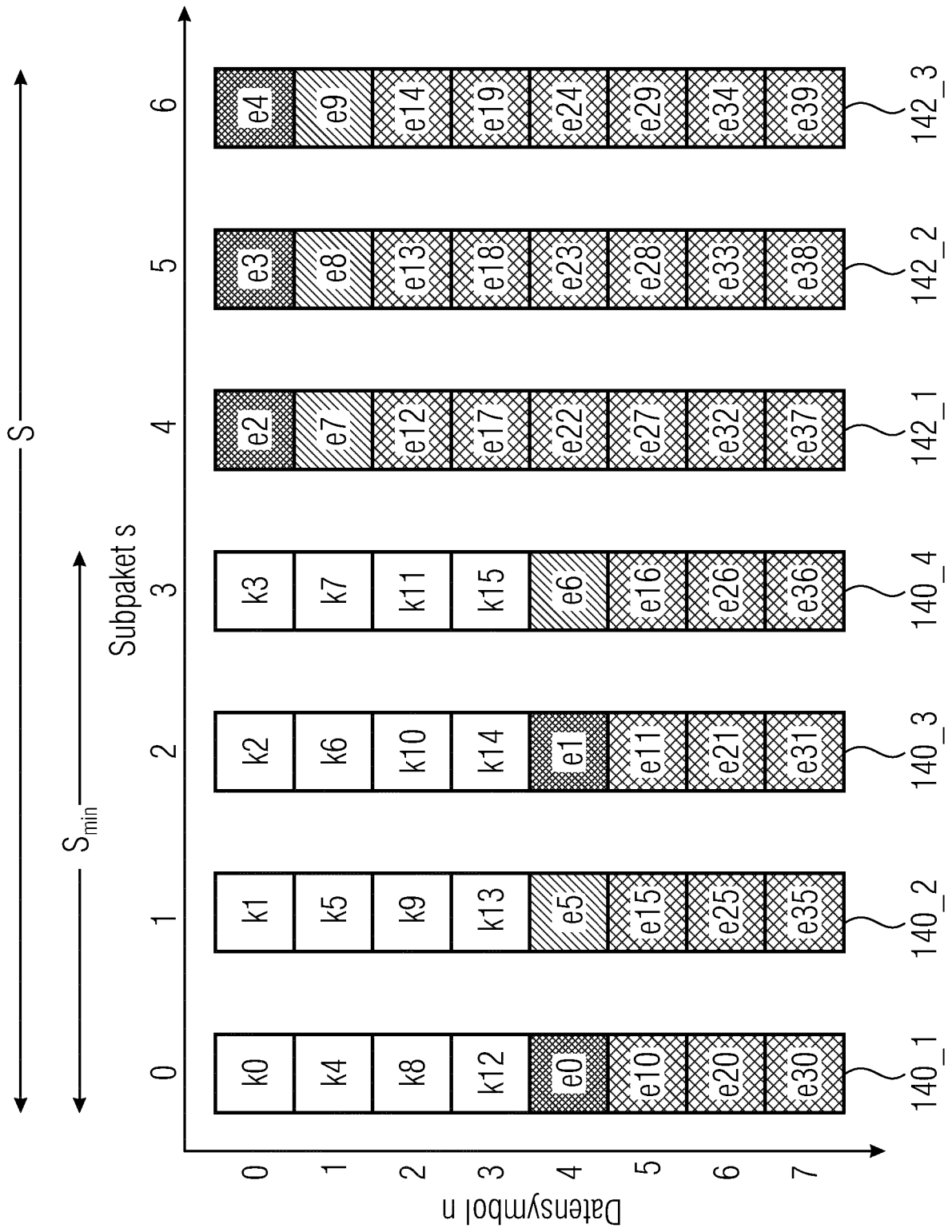


Fig. 6

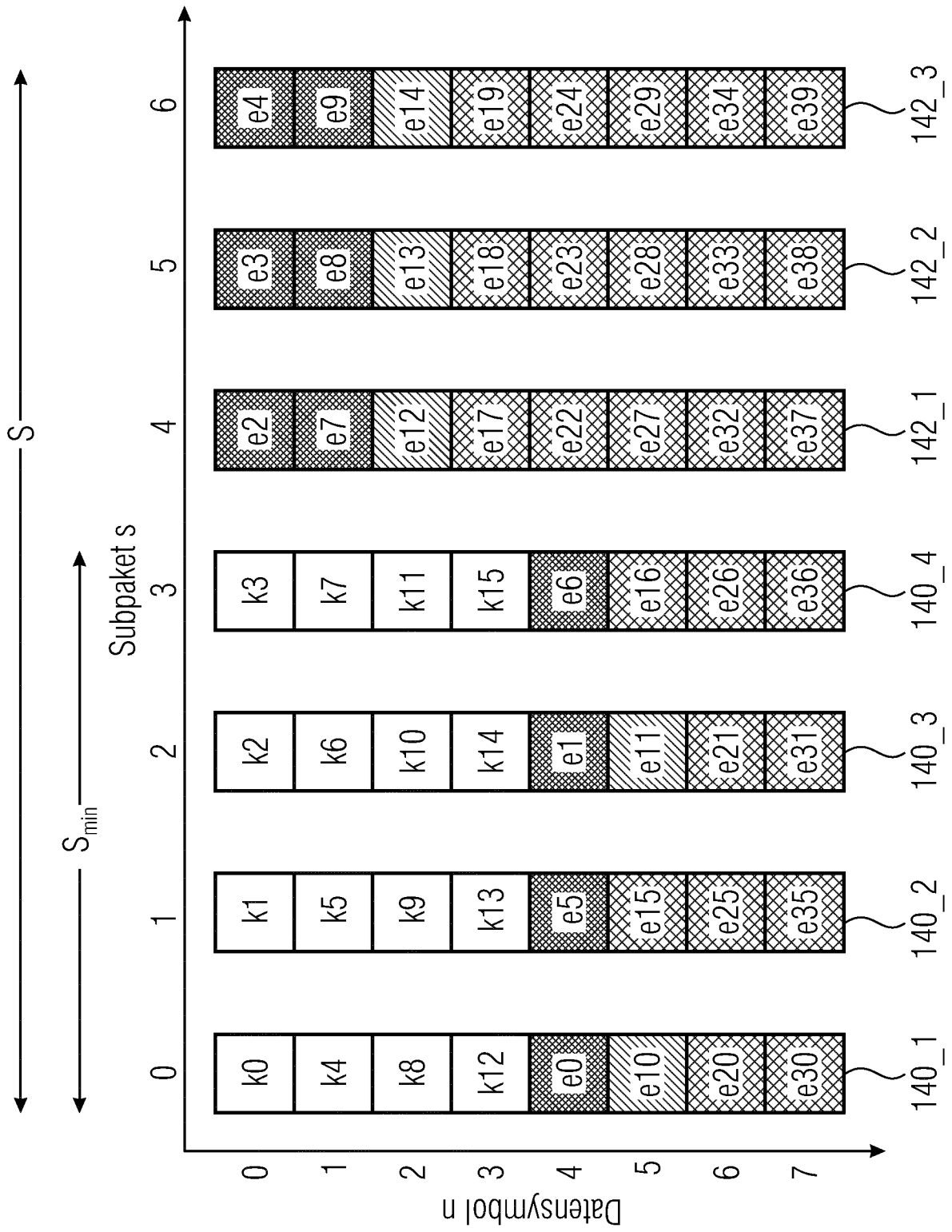


Fig. 7

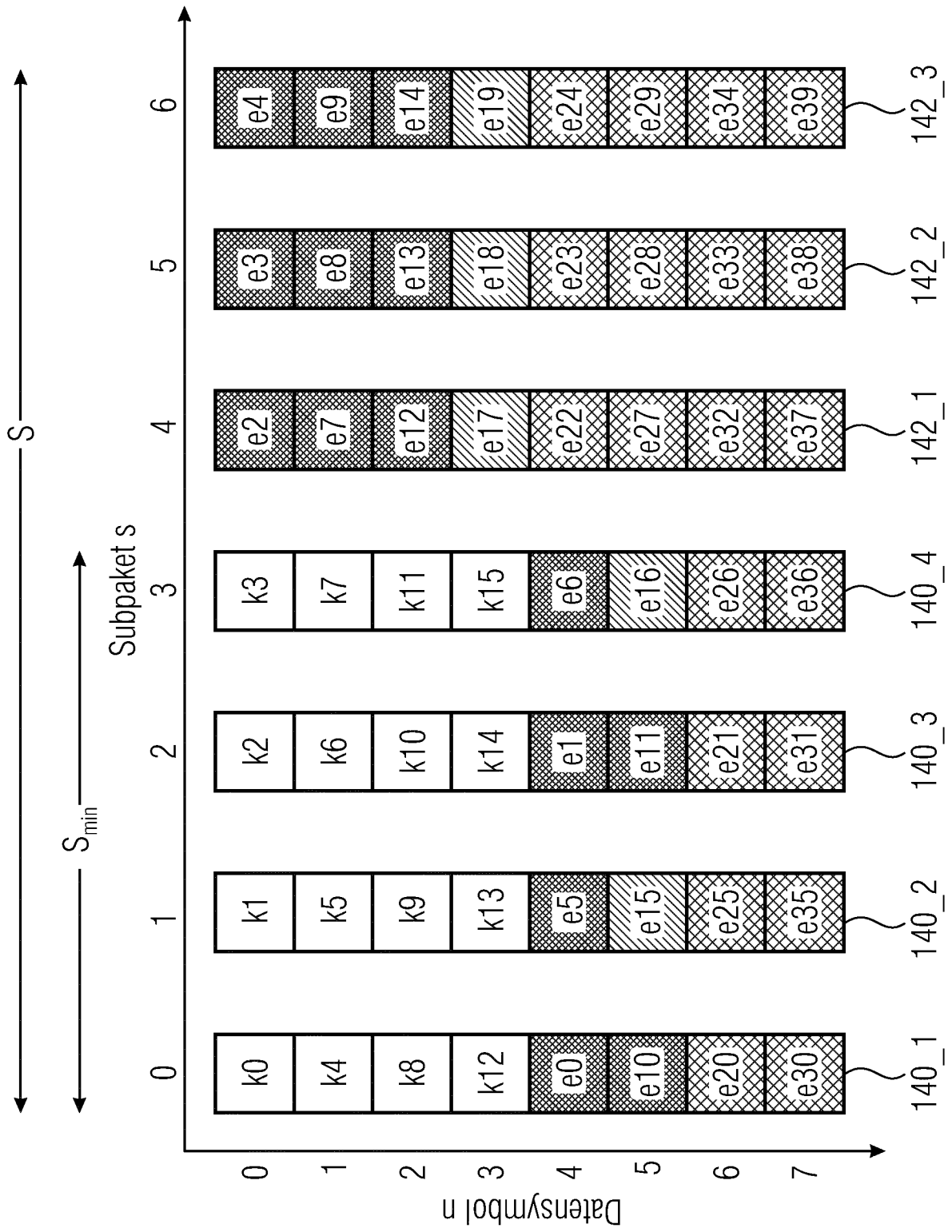


Fig. 8

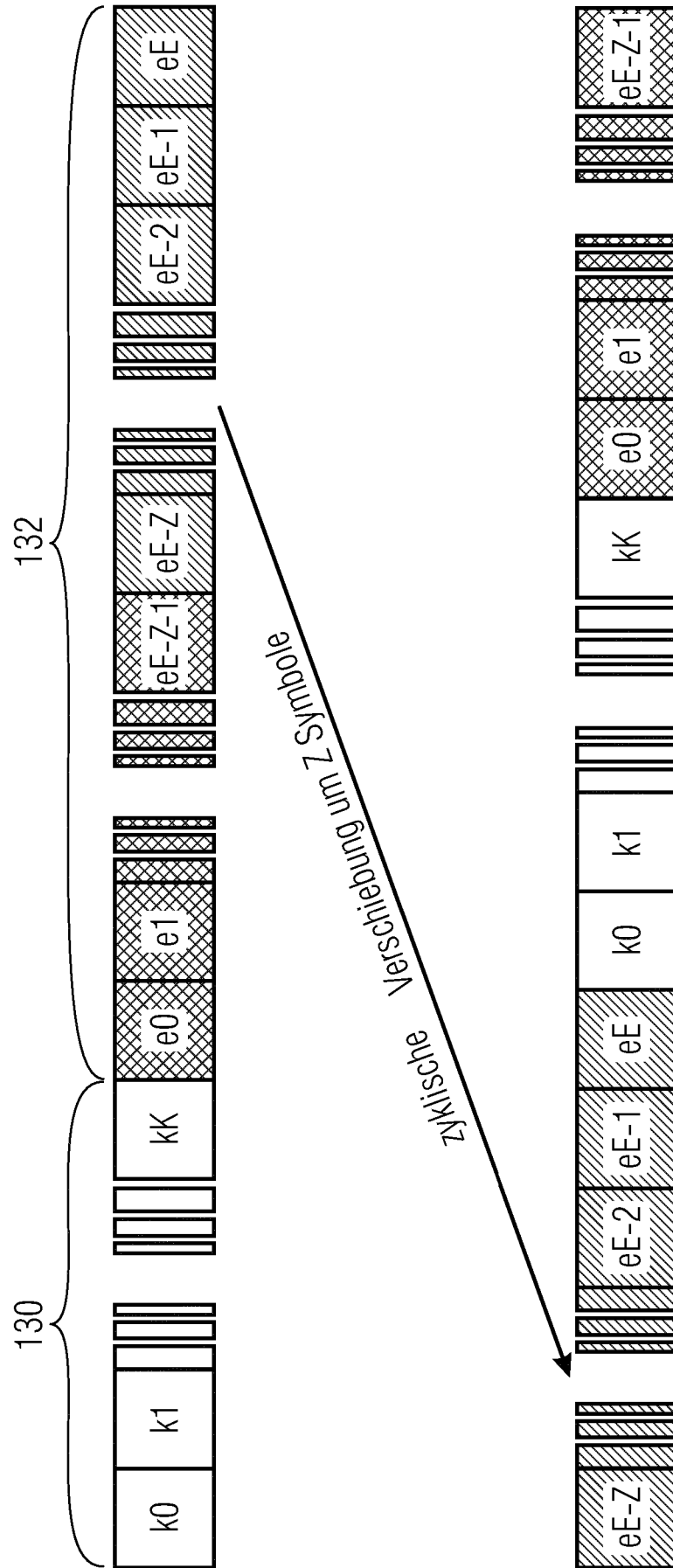


Fig. 9

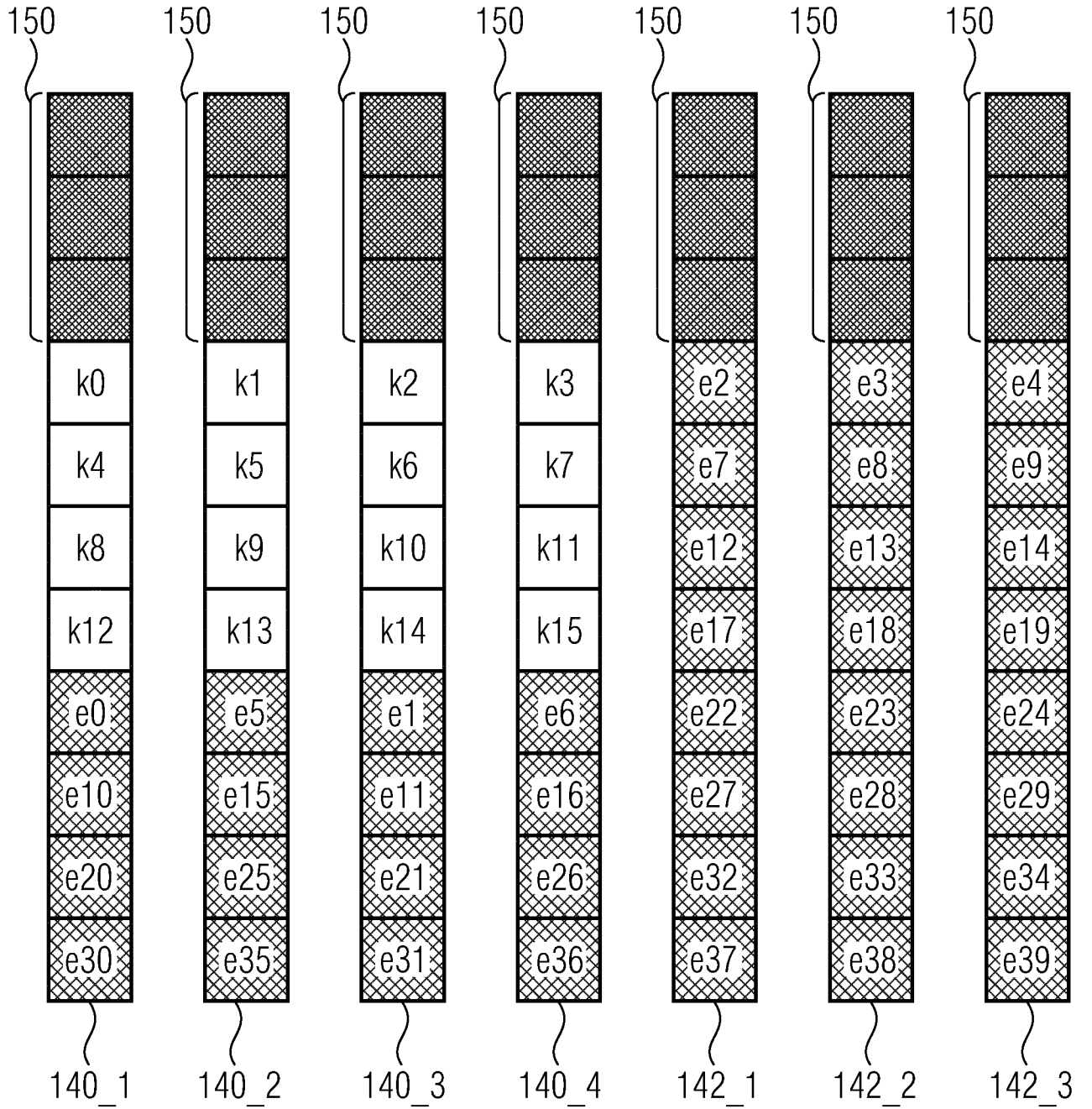


Fig. 10

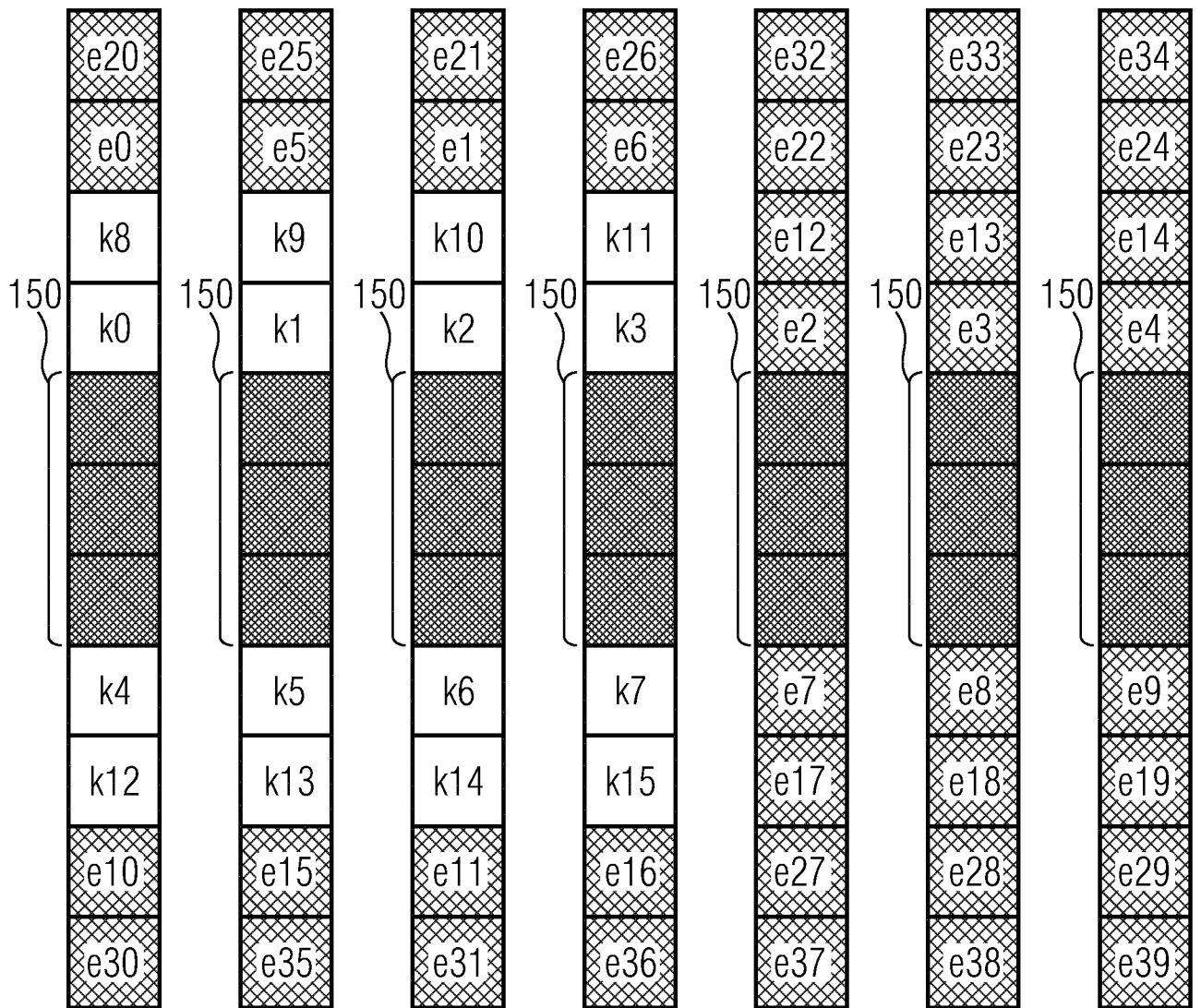


Fig. 11

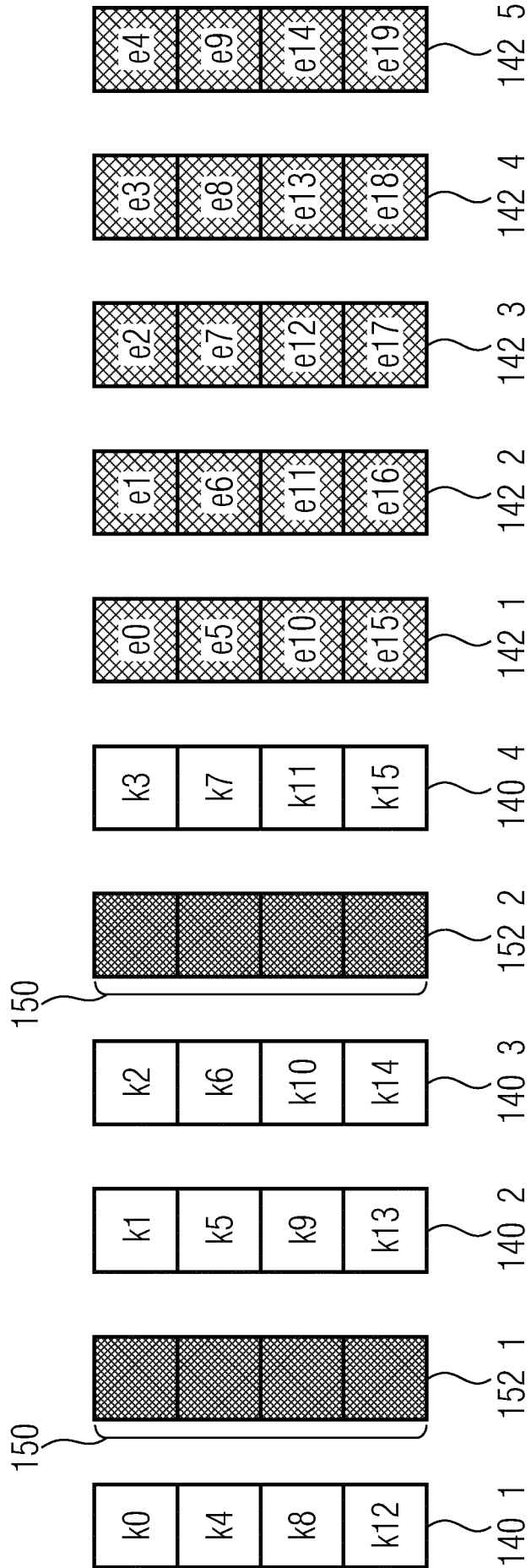


Fig. 12

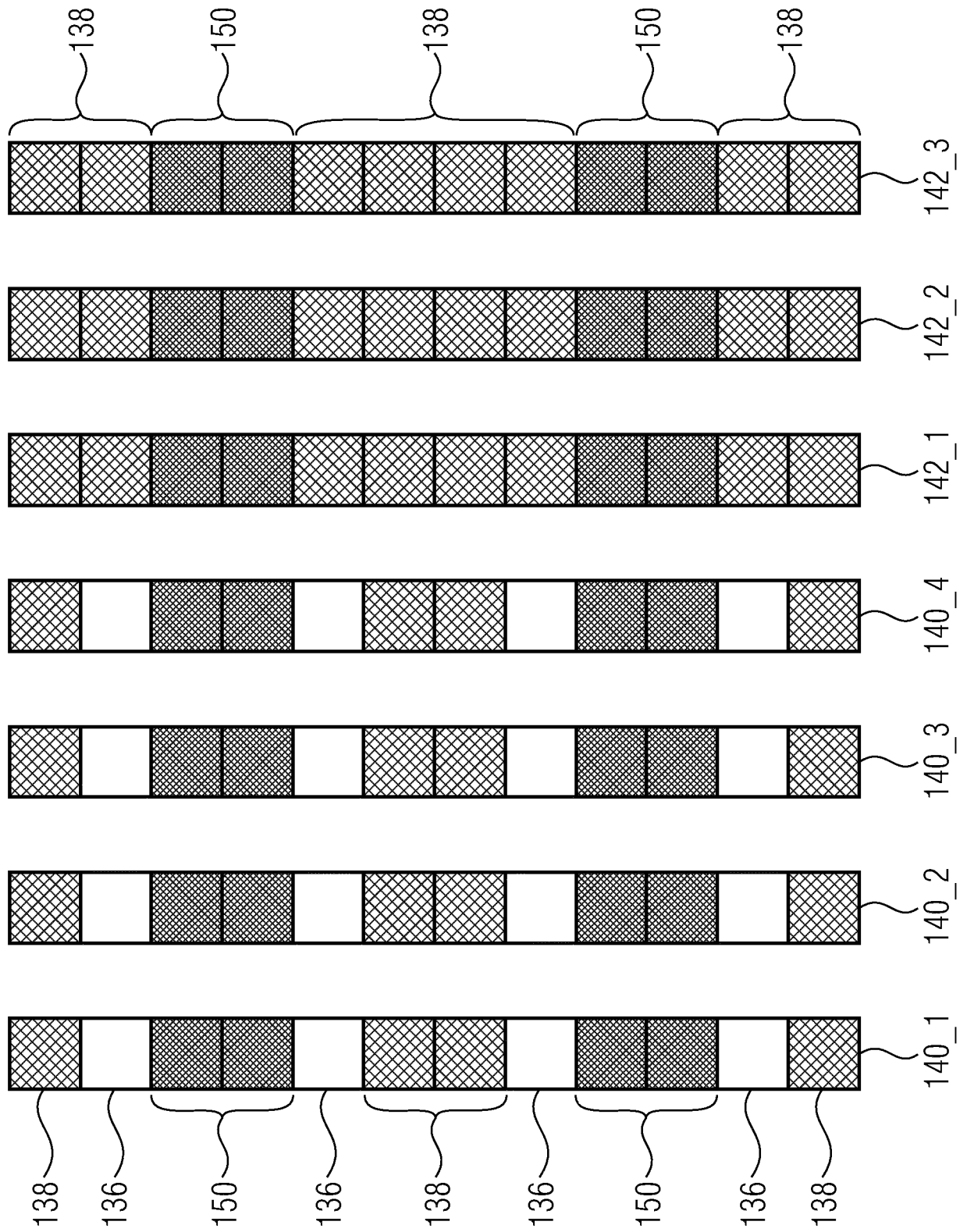


Fig. 13

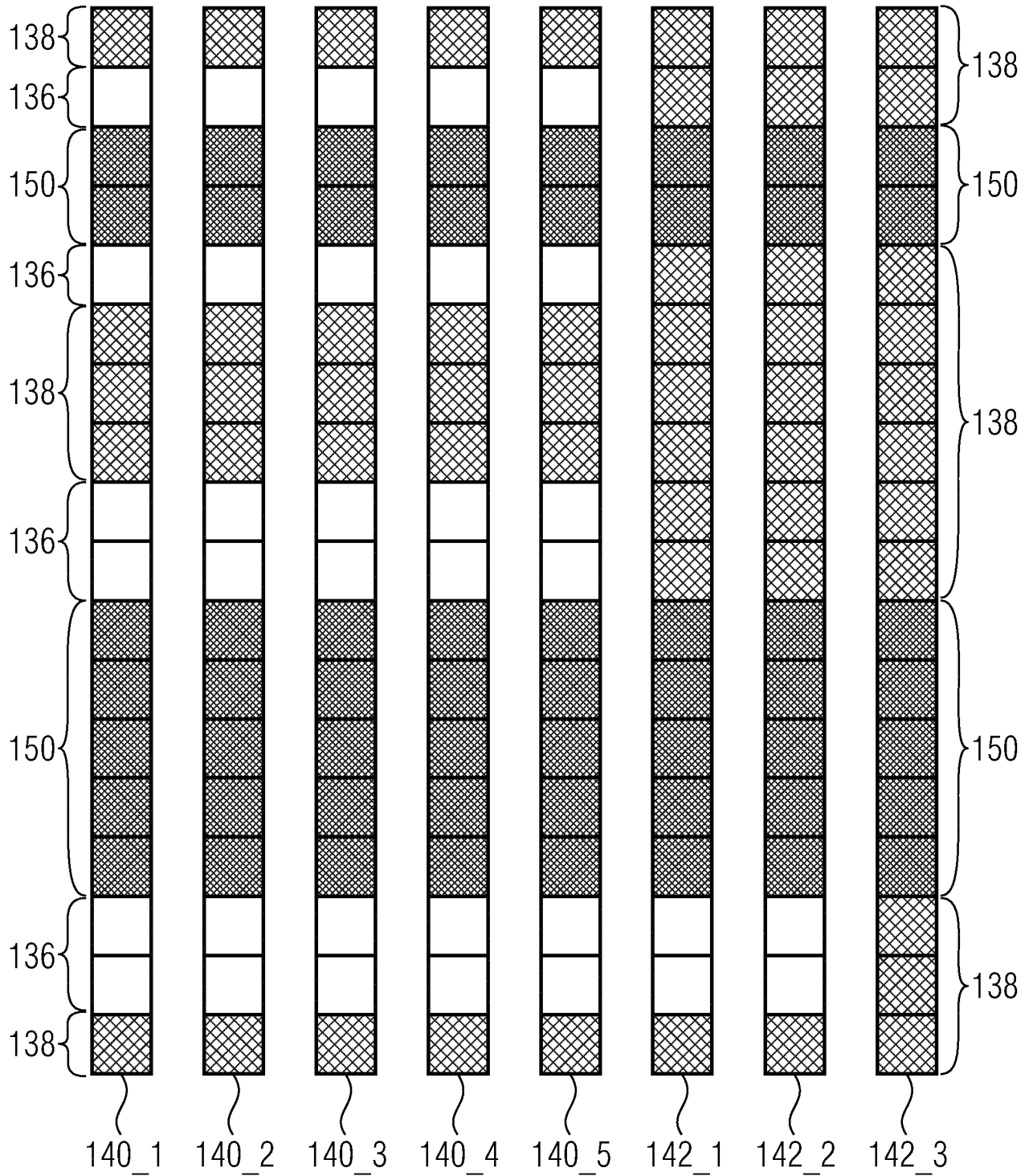


Fig. 14

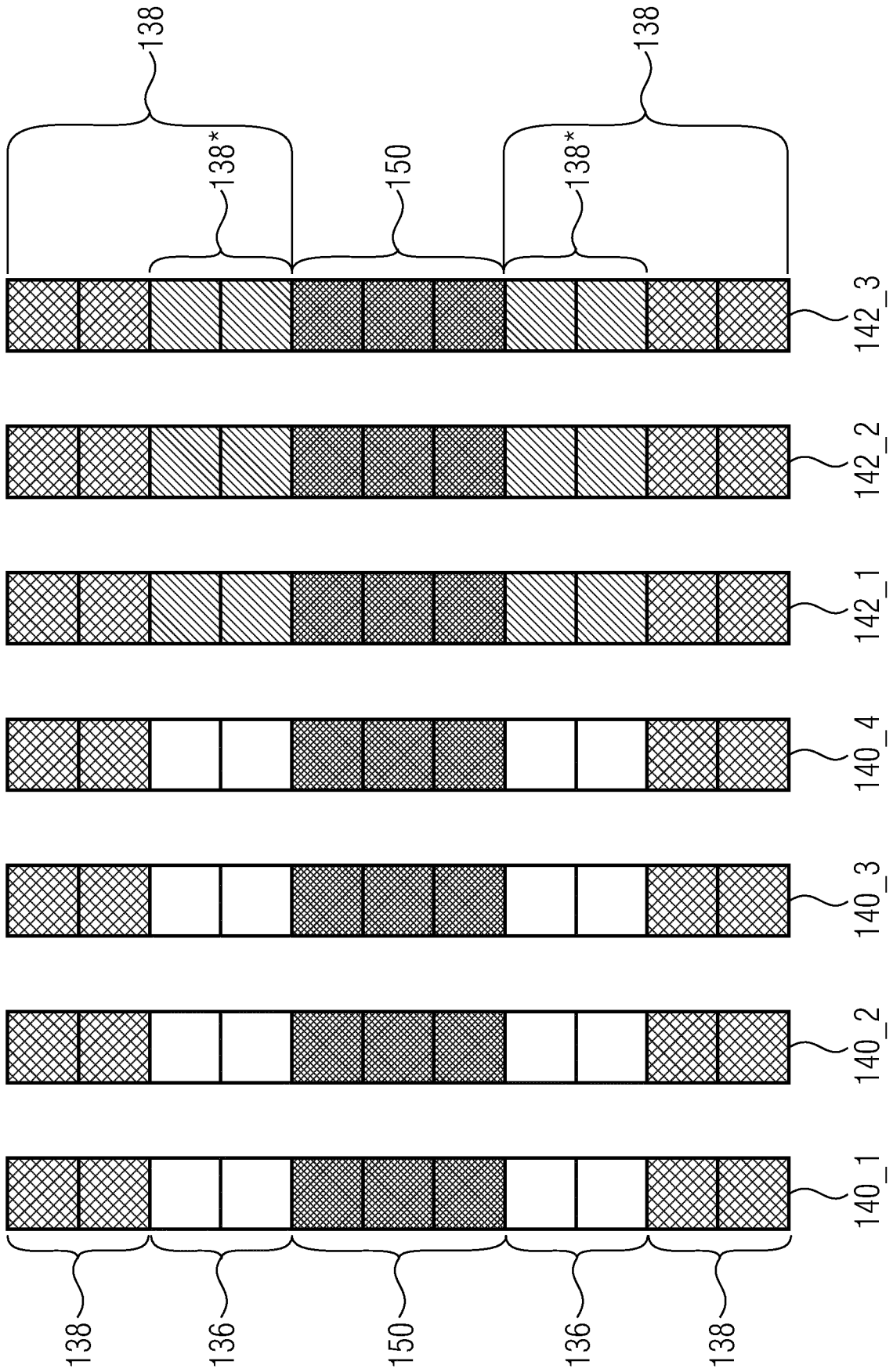


Fig. 15

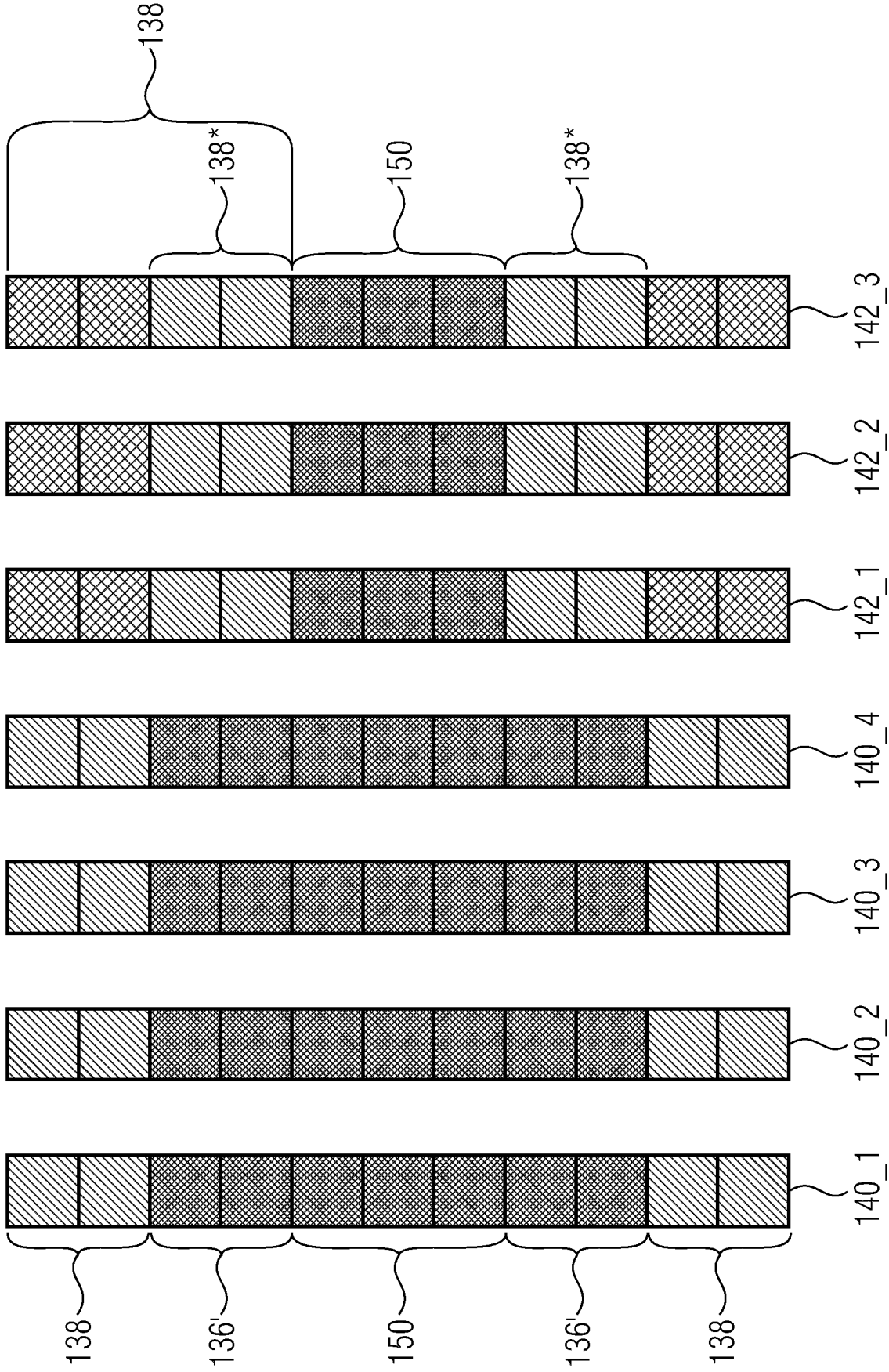


Fig. 16

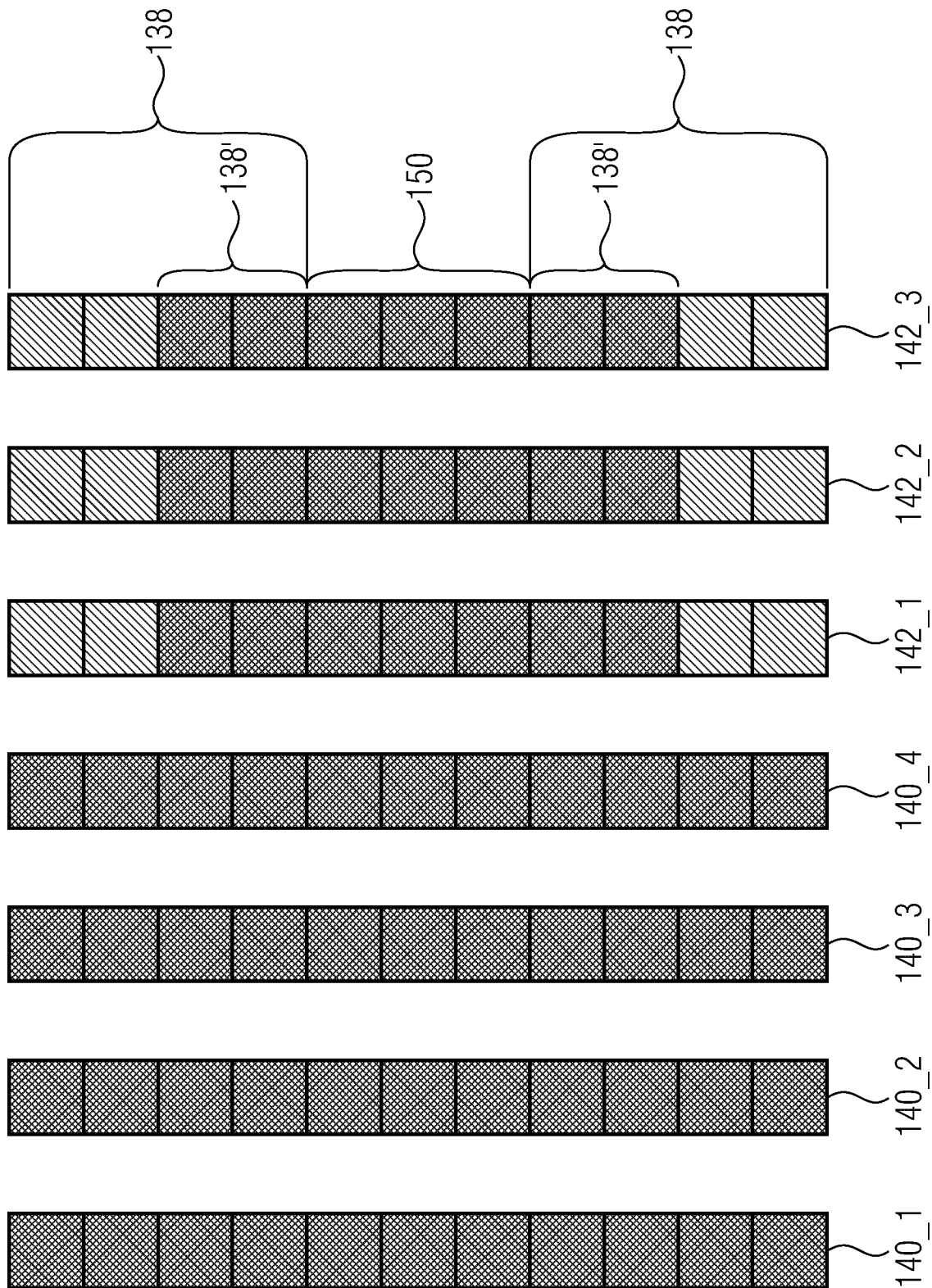


Fig. 17

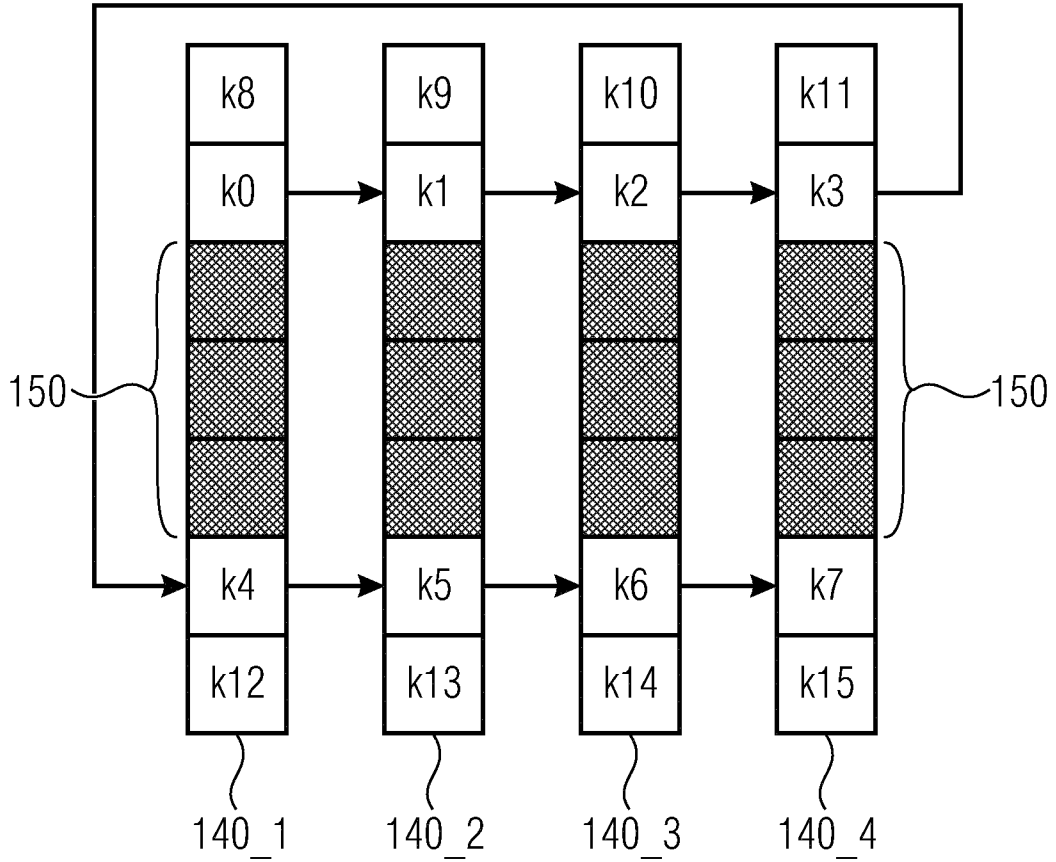


Fig. 18

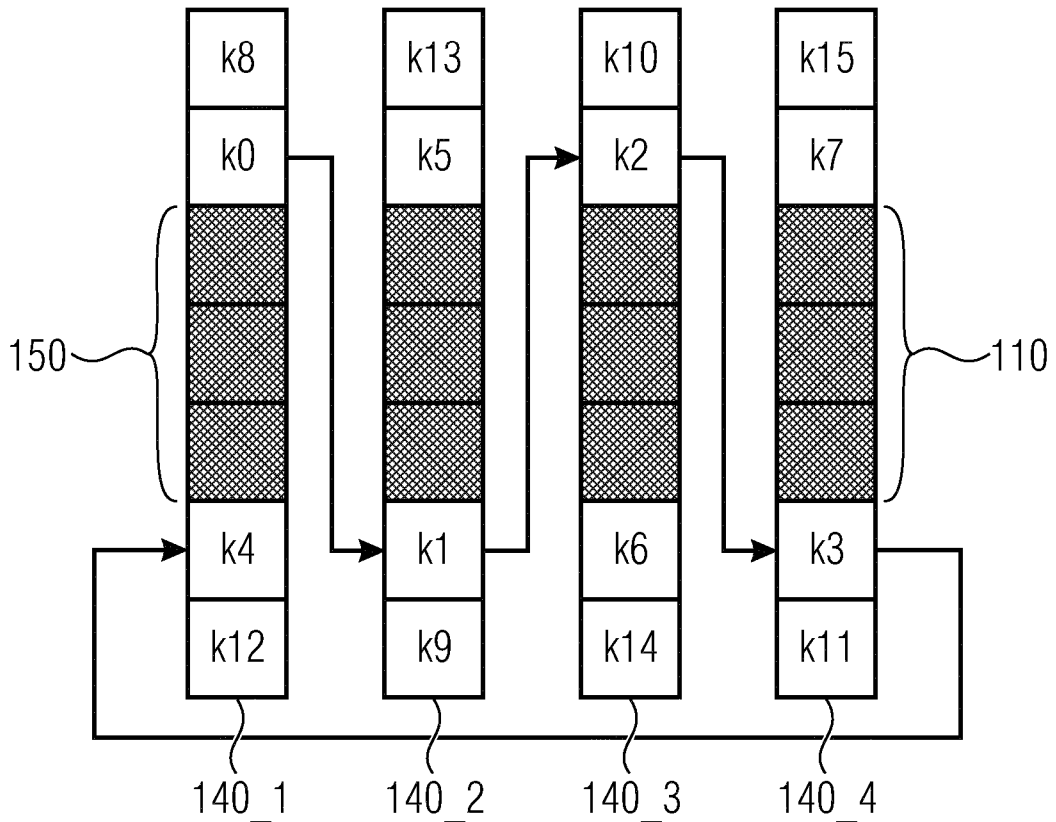


Fig. 19

19/19

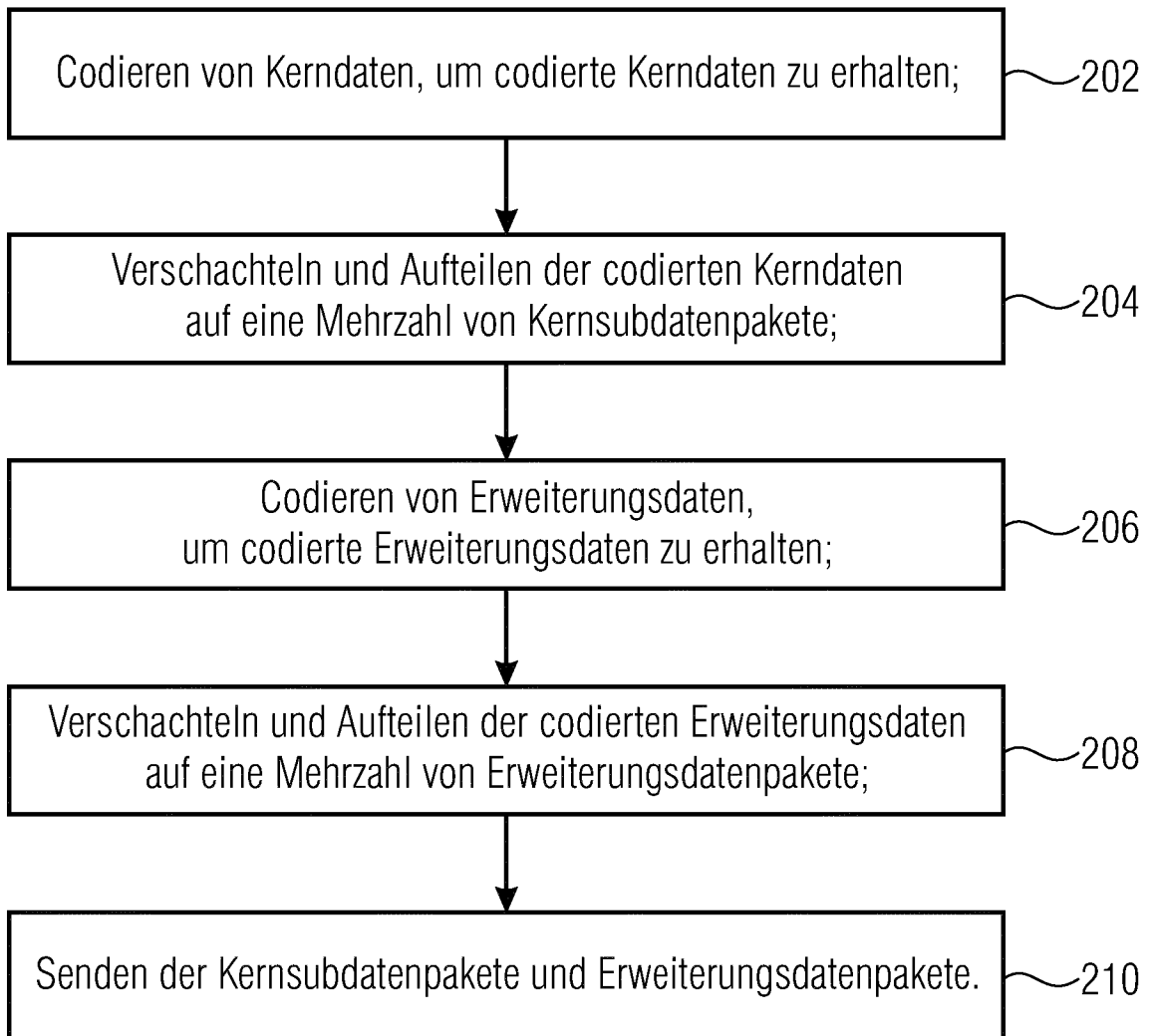
200

Fig. 20

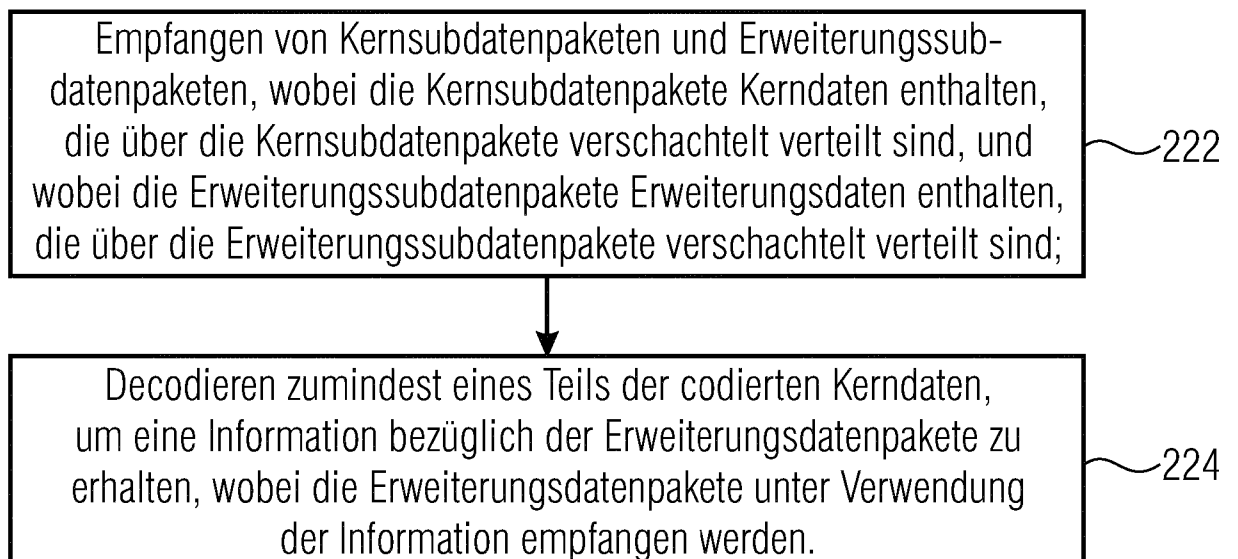
220

Fig. 21