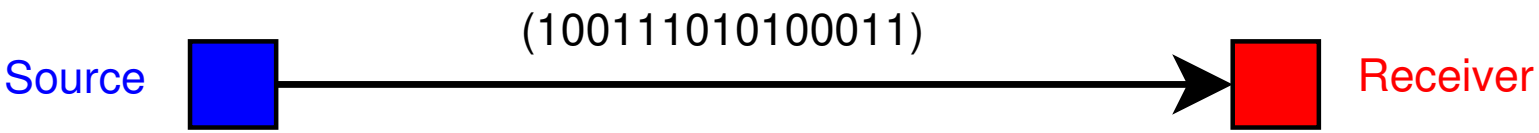
(Network) Coding und Verbindungen zur Systemtheorie

Anna-Lena Horlemann-Trautmann

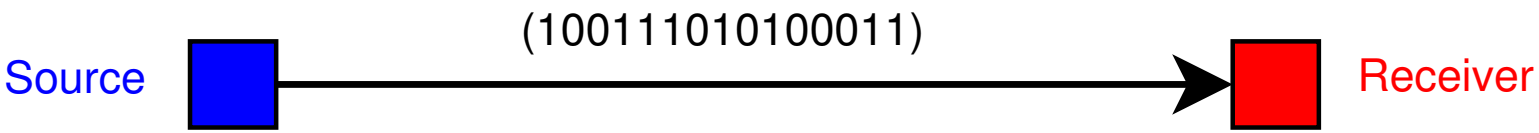
Algorithmics Laboratory, EPFL, Schweiz

10. Februar 2016
Elgersburg Workshop

Klassische Codierungstheorie:



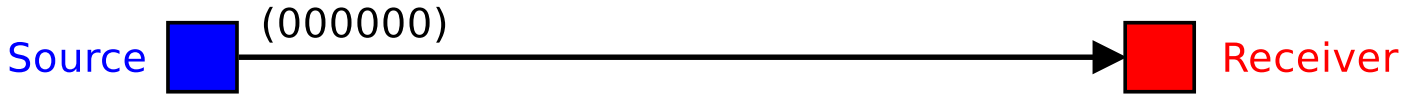
Klassische Codierungstheorie:



Was, wenn Fehler während der Übertragung passieren?

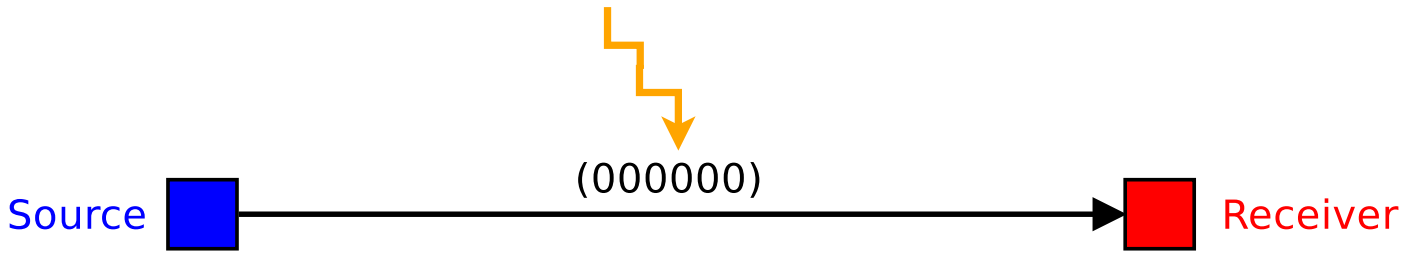
$\Longrightarrow$ Können korrigiert werden, indem man *Redundanz* den Daten hinzufügt, Daten *codiert*.

Einfachstes Beispiel – der Wiederholungs-Code:



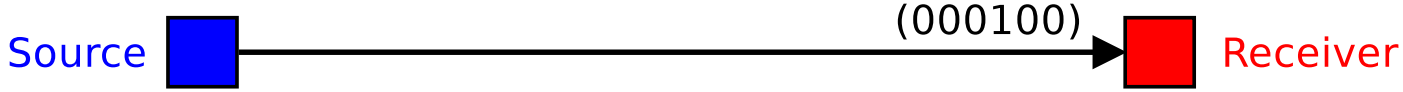
yes = (111111)
no = (000000)

Einfachstes Beispiel – der Wiederholungs-Code:



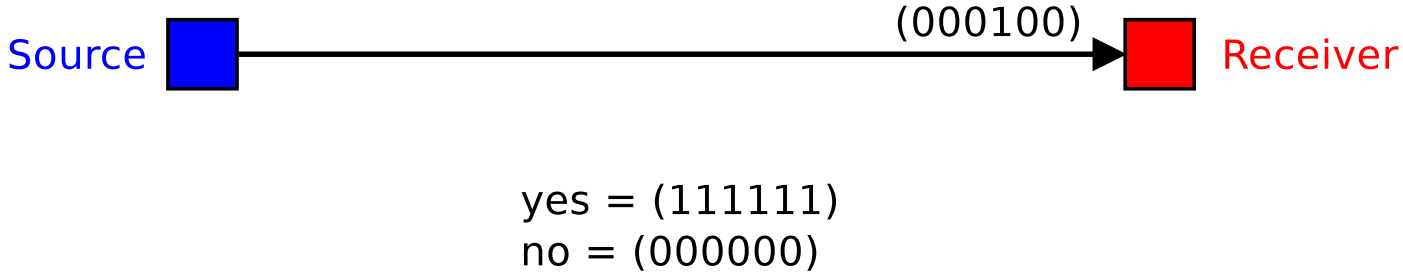
yes = (111111)
no = (000000)

Einfachstes Beispiel – der Wiederholungs-Code:



yes = (111111)
no = (000000)

Einfachstes Beispiel – der Wiederholungs-Code:



Da (000100) *näher* an (000000) als an (111111) ist, decodiert der Empfänger als (000000) und erhält so die Nachricht “no”.

Definition

Ein *Block-Code* über $\mathbb{F}_q$ der Länge n ist einfach eine Teilmenge von $\mathbb{F}_q^n$. Die Elemente des Codes nennt man *Codewörter*. Ein *linearer* Code ist ein Untervektorraum von $\mathbb{F}_q^n$.

Kanal-Modell

Sei $c \in \mathbb{F}_q^n$ ein über den Kanal gesendetes Codewort. Bei der Übertragung können additive Fehler vorkommen, dementsprechend ist das empfangen Wort

$$r = c + e$$

wobei $e \in \mathbb{F}_q^n$ der Fehlervektor ist.

Definition

Die Metrik, mit der die Anzahl der Fehler gemessen wird, ist die *Hamming-Metrik*:

$$d_H(r, c) := |\{i \mid r_i \neq c_i\}| \quad r, c \in \mathbb{F}_q^n$$

Definition

Die Metrik, mit der die Anzahl der Fehler gemessen wird, ist die *Hamming-Metrik*:

$$d_H(r, c) := |\{i \mid r_i \neq c_i\}| \quad r, c \in \mathbb{F}_q^n$$

Die *Minimal-Hammingdistanz* eines Codes C ist definiert als

$$d_{\min}(C) := \min\{d_H(b, c) \mid b, c \in C, b \neq c\}.$$

Definition

Die Metrik, mit der die Anzahl der Fehler gemessen wird, ist die *Hamming-Metrik*:

$$d_H(r, c) := |\{i \mid r_i \neq c_i\}| \quad r, c \in \mathbb{F}_q^n$$

Die *Minimal-Hammingdistanz* eines Codes C ist definiert als

$$d_{\min}(C) := \min\{d_H(b, c) \mid b, c \in C, b \neq c\}.$$

- Der Empfänger decodiert das empfangene Wort zum *nächsten* Codeword bzgl. der Hamming-Metrik.
- Je größer die Minimaldistanz des Codes, desto mehr Fehler können korrigiert werden.

Definition

Die Metrik, mit der die Anzahl der Fehler gemessen wird, ist die *Hamming-Metrik*:

$$d_H(r, c) := |\{i \mid r_i \neq c_i\}| \quad r, c \in \mathbb{F}_q^n$$

Die *Minimal-Hammingdistanz* eines Codes C ist definiert als

$$d_{\min}(C) := \min\{d_H(b, c) \mid b, c \in C, b \neq c\}.$$

- Der Empfänger decodiert das empfangene Wort zum *nächsten* Codeword bzgl. der Hamming-Metrik.
- Je größer die Minimaldistanz des Codes, desto mehr Fehler können korrigiert werden.
- Tradeoff: Je mehr Redundanz hinzugefügt wird, desto schlechter wird die Übertragungsrate.

Beispiel

Über $\mathbb{F}_3 = \{0, 1, 2\}$ definieren wir den linearen Code C der Länge 4 und Dimension 2 mit der *Generatormatrix*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 1 \end{pmatrix}.$$

Der Code hat $3^2 = 9$ Codewörter und Minimal-Hammingdistanz 2.

Beispiel

Über $\mathbb{F}_3 = \{0, 1, 2\}$ definieren wir den linearen Code C der Länge 4 und Dimension 2 mit der *Generatormatrix*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 1 \end{pmatrix}.$$

Der Code hat $3^2 = 9$ Codewörter und Minimal-Hammingdistanz 2. Äquivalent können wir C als Kern der *Kontrollmatrix*

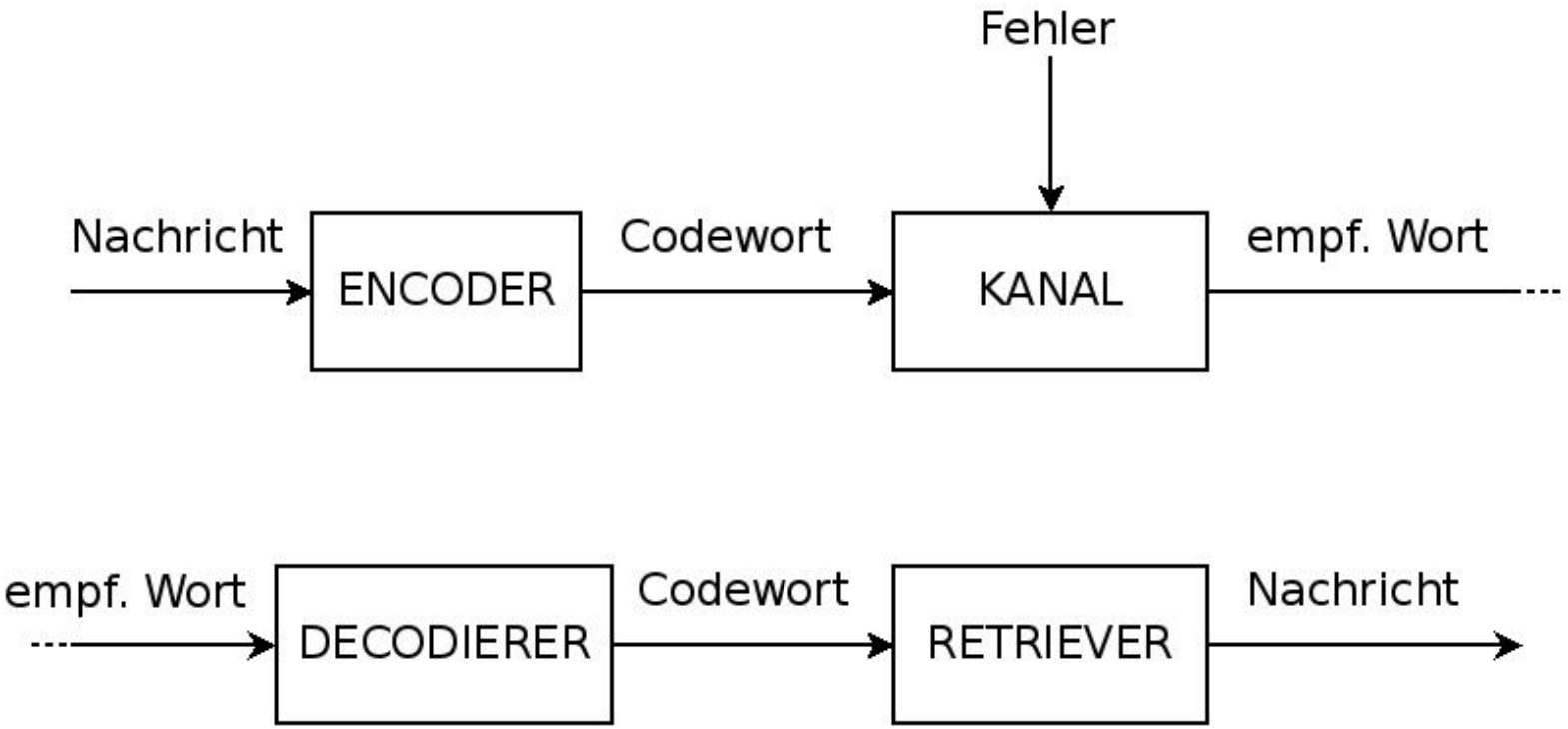
$$H = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{pmatrix}$$

definieren (mit $GH^T = 0$).

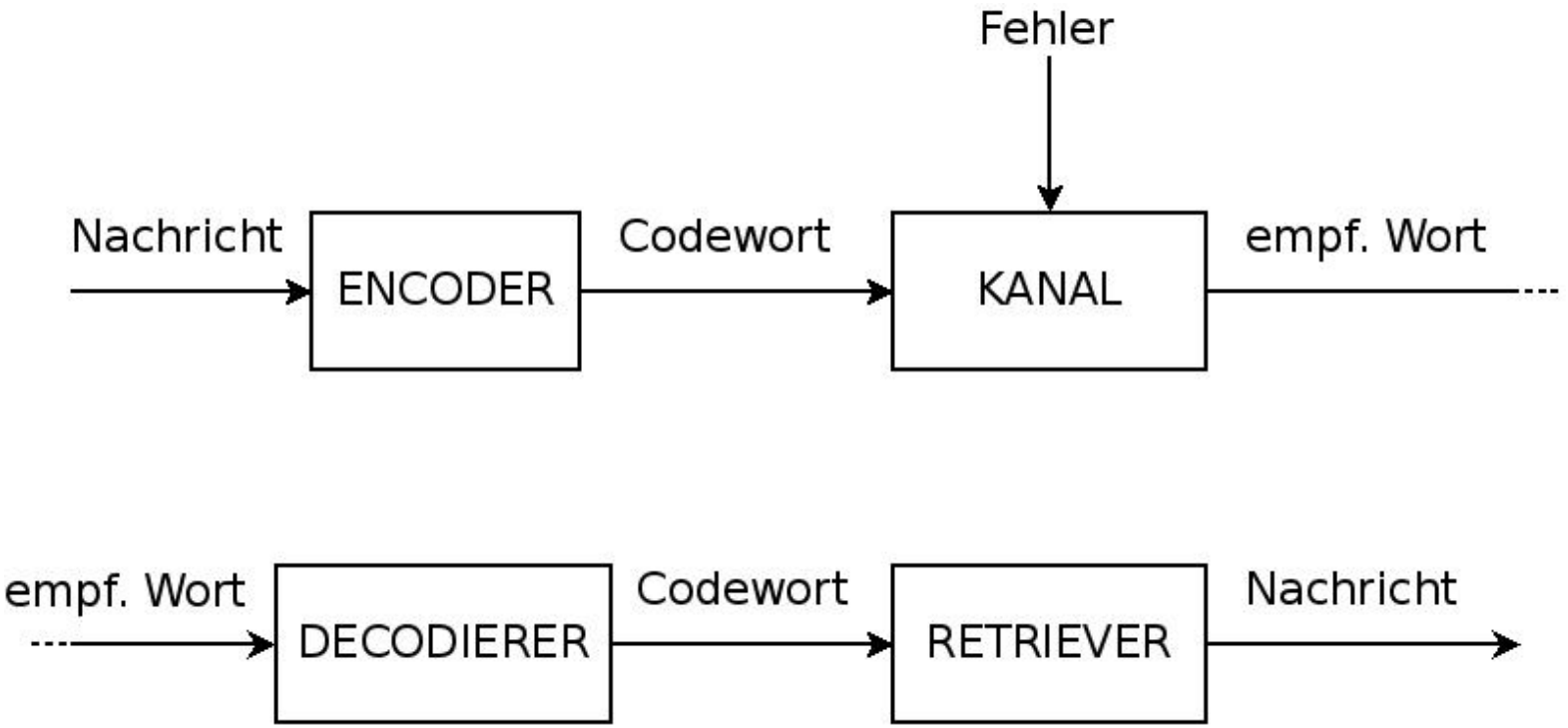
Hauptforschungsziele (Block-Codes):

1. Für einen gegebenen Raum $\mathbb{F}_q^n$ und Minimaldistanz, finde die besten Packungen bzgl. der Hamming-Metrik.
   $\Longrightarrow$ beste mögliche Übertragungsrate bei gleicher Fehlerkorrektur
2. Finde effiziente Decodieralgorithmen (möglicherweise zusammen mit Code-Konstruktionen).
   $\Longrightarrow$ schnellere Kommunikation

Der komplette Vorgang:



Der komplette Vorgang:



Manchmal können Decodierer und Retriever auch zusammengelegt werden.

Systeme:

	Input	**Output**
Encoder	Nachricht	Codewort
Kanal	Codewort	empfangenes Wort
Decodierer	empfangenes Wort	Codewort
Retriever	Codeword	Nachricht

Systeme:

	Input	**Output**
Encoder	Nachricht	Codewort
Kanal	Codewort	empfangenes Wort
Decodierer	empfangenes Wort	Codewort
Retriever	Codeword	Nachricht

- Diese Systeme sind alle diskret.
- Als Encoder wird oft eine lineare Abbildung von $\mathbb{F}_q^k$ nach $\mathbb{F}_q^n$ genommen.
- Der Retriever ist die inverse Abbildung vom Encoder und daher auch linear.
- Kanal ist stochastisches System (Fehler folgt Wahrscheinlichkeitsverteilung).

Systeme:

	Input	**Output**
Encoder	Nachricht	Codewort
Kanal	Codewort	empfangenes Wort
Decodierer	empfangenes Wort	Codewort
Retriever	Codeword	Nachricht

- Diese Systeme sind alle diskret.
- Als Encoder wird oft eine lineare Abbildung von $\mathbb{F}_q^k$ nach $\mathbb{F}_q^n$ genommen.
- Der Retriever ist die inverse Abbildung vom Encoder und daher auch linear.
- Kanal ist stochastisches System (Fehler folgt Wahrscheinlichkeitsverteilung).

$\Longrightarrow$ Interessant: Decodierer

Decodierer + Retriever:

$$\begin{aligned}\text{Nachricht} &= \text{argmin}\{d_H(mG, r) \mid m \in \mathbb{F}_q^k\} \\ &= \text{argmin}\{||mG - r||_H \mid m \in \mathbb{F}_q^k\}\end{aligned}$$

$G \in \mathbb{F}_q^{k \times n}$ Generatormatrix
$r \in \mathbb{F}_q^n$ empfangenes Wort

Verschiedene Dekodieralgorithmen:

- Mit der Hilfe der Kontrollmatrix (bzw. Syndromen), z.B. Berlekamp-Massey-Algorithmus.
- Mit Polynom-Interpolation, z.B. Welch-Berlekamp-Algorithmus.
- Mit Belief-Propagation-Algorithmus.
- ...

Idee von Berlekamp-Massey-Algorithmus:

- Sei $\alpha \in \mathbb{F}_q$ Einheitswurzel und $C \subseteq \mathbb{F}_q^n$ ein BCH-Code mit Kontrollmatrix

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^n \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2n} \\ \vdots & & & & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \dots & \alpha^{n(n-k)} \end{pmatrix}$$

und sei $r = c + e$ das empfangene Wort mit

$$e = (0...0e_{i_1}...0...e_{i_2}...0...e_{i_t}0...0).$$

- Berechne die *Syndrome*

$$(s_1, s_2, \dots, s_{n-k}) = rH^T = cH^T + eH^T = eH^T.$$

Idee von Berlekamp-Massey-Algorithmus:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^n \\ & & \vdots & & \end{pmatrix}$$

- Definiere die *Error-Locators* $X_\ell := \alpha^{-i_\ell}$ und das *Error-Locator-Polynom*

$$\Lambda(z) = \sum_{i=0}^{t} \Lambda_i z^i := \prod_{z=1}^{t} (z - X_\ell).$$

- Mit Newton-Gleichungen bekommt man

$$s_{i+t+1} + \Lambda_1 s_{i+t} + \cdots + \Lambda_t s_{i+1} = 0$$

für $i = 1, \dots, n - k - t + 1$.

Berlekamp-Massey-Algorithmus:

- Finde Polynom $\Lambda(x) = \sum \Lambda_i x^i$ vom kleinsten Grad t, s.d.

$$s_{i+t+1} + \Lambda_1 s_{i+t} + \cdots + \Lambda_t s_{i+1} = 0$$

  für $i = 0, \ldots, n-k-t$.
- Äquivalent: Finde das kürzeste linear rückgekoppelte Schieberegister, so dass der Output $s_1, s_2, \ldots, s_{n-k}$ ist. ($\Lambda(D)$ ist Rückkopplungspolynom des Schieberegisters.)

Berlekamp-Massey-Algorithmus:

- Finde Polynom $\Lambda(x) = \sum \Lambda_i x^i$ vom kleinsten Grad t, s.d.

$$s_{i+t+1} + \Lambda_1 s_{i+t} + \cdots + \Lambda_t s_{i+1} = 0$$

  für $i = 0, \ldots, n-k-t$.
- Äquivalent: Finde das kürzeste linear rückgekoppelte Schieberegister, so dass der Output $s_1, s_2, \ldots, s_{n-k}$ ist. ($\Lambda(D)$ ist Rückkopplungspolynom des Schieberegisters.)

$\Longrightarrow$ *irreduzible Transferfunktion / minimale Realisierung*

Berlekamp-Massey-Algorithmus:

- Finde Polynom $\Lambda(x) = \sum \Lambda_i x^i$ vom kleinsten Grad t, s.d.

$$s_{i+t+1} + \Lambda_1 s_{i+t} + \cdots + \Lambda_t s_{i+1} = 0$$

  für $i = 0, \ldots, n-k-t$.

- Äquivalent: Finde das kürzeste linear rückgekoppelte Schieberegister, so dass der Output $s_1, s_2, \ldots, s_{n-k}$ ist. ($\Lambda(D)$ ist Rückkopplungspolynom des Schieberegisters.)

$\Longrightarrow$ *irreduzible Transferfunktion / minimale Realisierung*

- Kann mit iterativem Algorithmus effizient berechnet werden (siehe Wikipedia o.ä.).

Berlekamp-Massey-Algorithmus:

- Finde Nullstellen $-i_1, \ldots, -i_t$ von $\Lambda(x)$ ($\mod q-1$).
- Mit Kenntnis der Fehlerpositionen $i_1, i_2, \ldots, i_t$ können wir dann die Fehlerwerte e_{i_j} finden, indem wir das lineare Gleichungssystem

$$(e_{i_1}, \ldots, e_{i_t}) \begin{pmatrix} \alpha^{i_1} & \alpha^{2i_1} & \ldots & \alpha^{(n-k)i_1} \\ \alpha^{i_2} & \alpha^{2i_2} & \ldots & \alpha^{(n-k)i_2} \\ & & \vdots & \\ \alpha^{i_t} & \alpha^{2i_t} & \ldots & \alpha^{(n-k)i_t} \end{pmatrix} = (s_1, \ldots, s_{n-k})$$

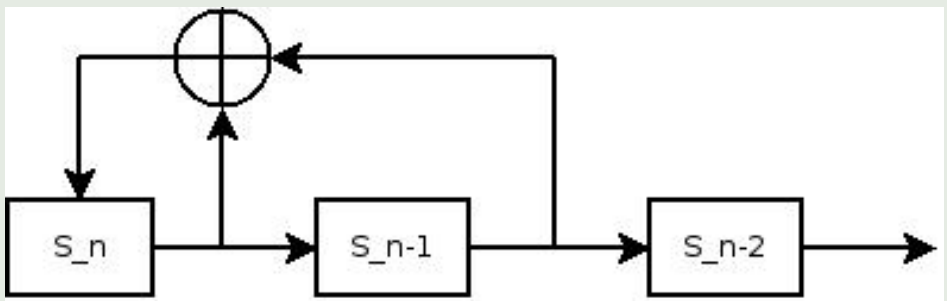
  lösen.
- Dann $e = (0 \ldots 0 e_{i_1} \ldots 0 \ldots e_{i_2} \ldots 0 \ldots e_{i_t} 0 \ldots 0)$ und $c = r - e$.

Beispiel

Sei $\alpha \in \mathbb{F}_{2^4}$ primitive Einheitswurzel mit $\alpha^4 = \alpha + 1$. Wir betrachten $\mathbb{F}_{2^4} = \mathbb{F}_2[\alpha]$ und den BCH-Code der Länge $n = 11$. Für ein empfangenes r berechnen wir die Syndromfolge

$$(s_1, s_2, \ldots, s_7) = (1, 1, 0, 1, 1, 0, 1).$$

Der Berlekamp-Massey-Algorithmus ermittelt $\Lambda(x) = x^2 + x + 1$, d.h. das Schieberegister ist von der Form



.

Beispiel fortgesetzt

Wir faktorisieren

$$\Lambda(x) = x^2 + x + 1 = (x - \alpha^5)(x - \alpha^{10}).$$

Also Fehler in Positionen $i_2 = -5 \equiv 10$ und $i_1 = -10 \equiv 5$.

Beispiel fortgesetzt

Wir faktorisieren

$$\Lambda(x) = x^2 + x + 1 = (x - \alpha^5)(x - \alpha^{10}).$$

Also Fehler in Positionen $i_2 = -5 \equiv 10$ und $i_1 = -10 \equiv 5$.
Wir lösen

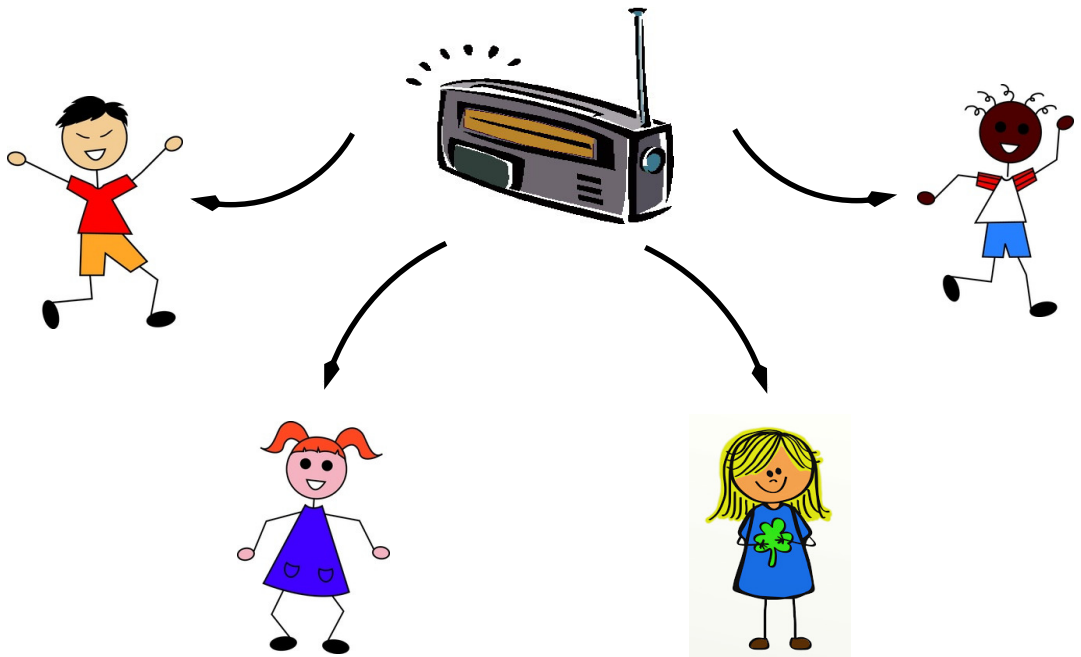
$$(e_5, e_{10}) \begin{pmatrix} \alpha^5 & \alpha^{10} & \dots & \alpha^{35} \\ \alpha^{10} & \alpha^{20} & \dots & \alpha^{70} \end{pmatrix} = (1101101)$$

$$\iff (e_5, e_{10}) = (1, 1).$$

Also Fehlervektor $e = (00000100001)$ und $c = r - e$.

Network Coding

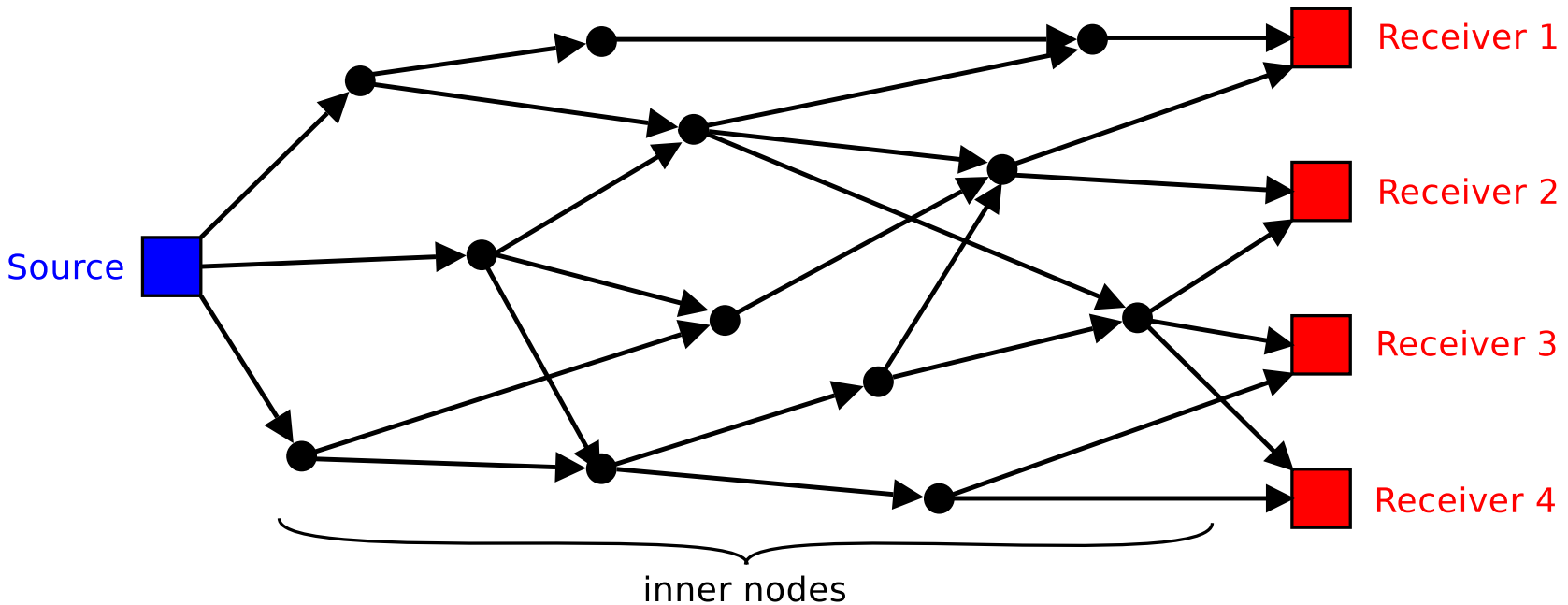
- Das Multicast-Modell:



Alle Empfänger sollen (gleichzeitig) die selben Daten erhalten.

Network Coding

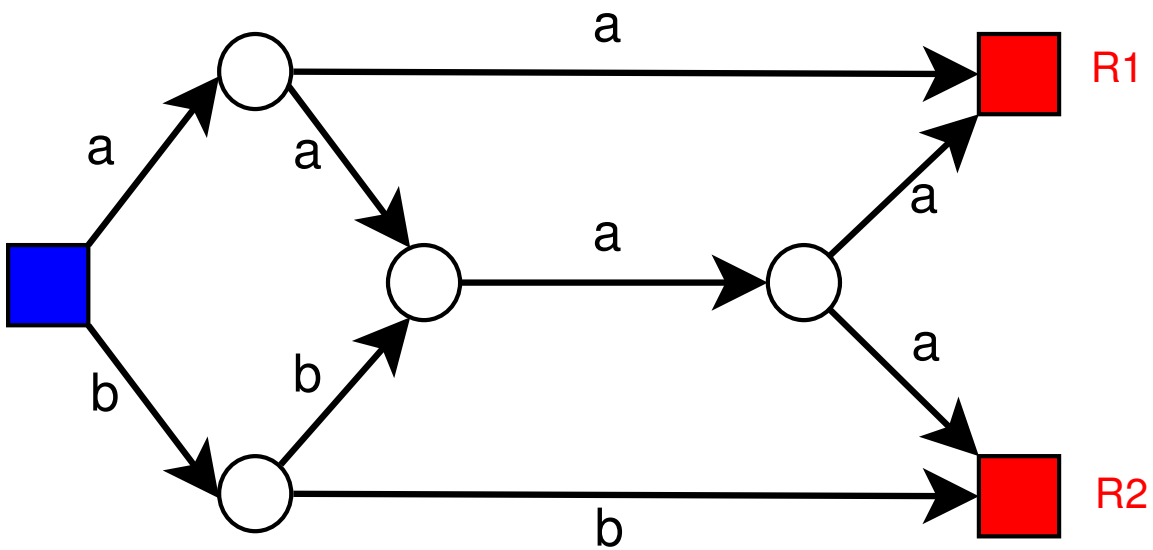
- Das Multicast-Modell:



Alle Empfänger sollen (gleichzeitig) die selben Daten erhalten.

“Codieren ist besser als Weiterleiten”

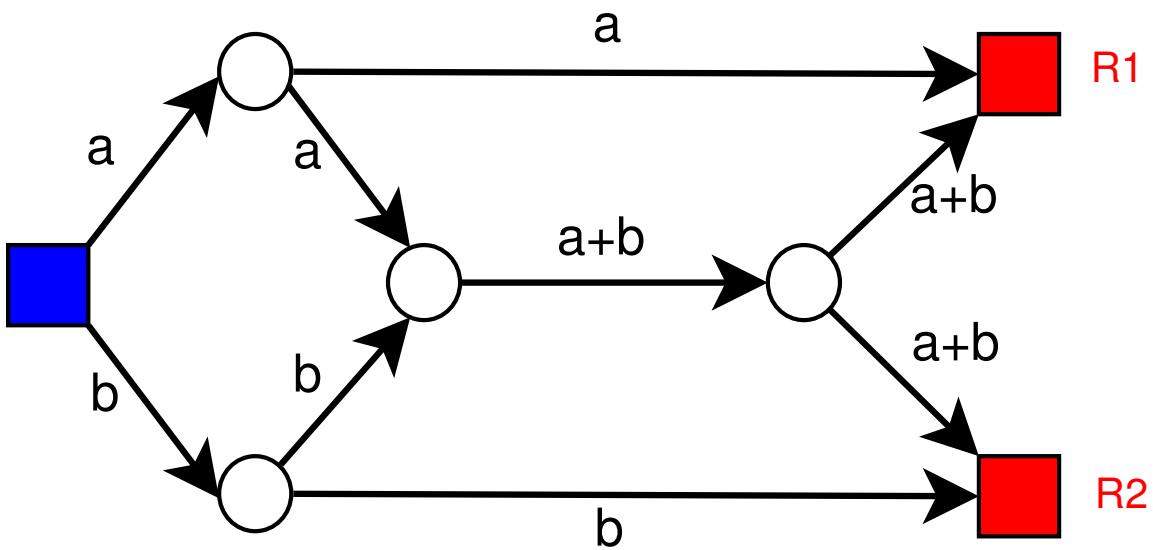
Das Schmetterling-Netzwerk mit **Weiterleiten**:



R2 erhält a und b, aber R1 erhält nur a.
$\Longrightarrow$ 2 Zeiteinheiten benötigt, um a, b an beide Empfänger zu senden.

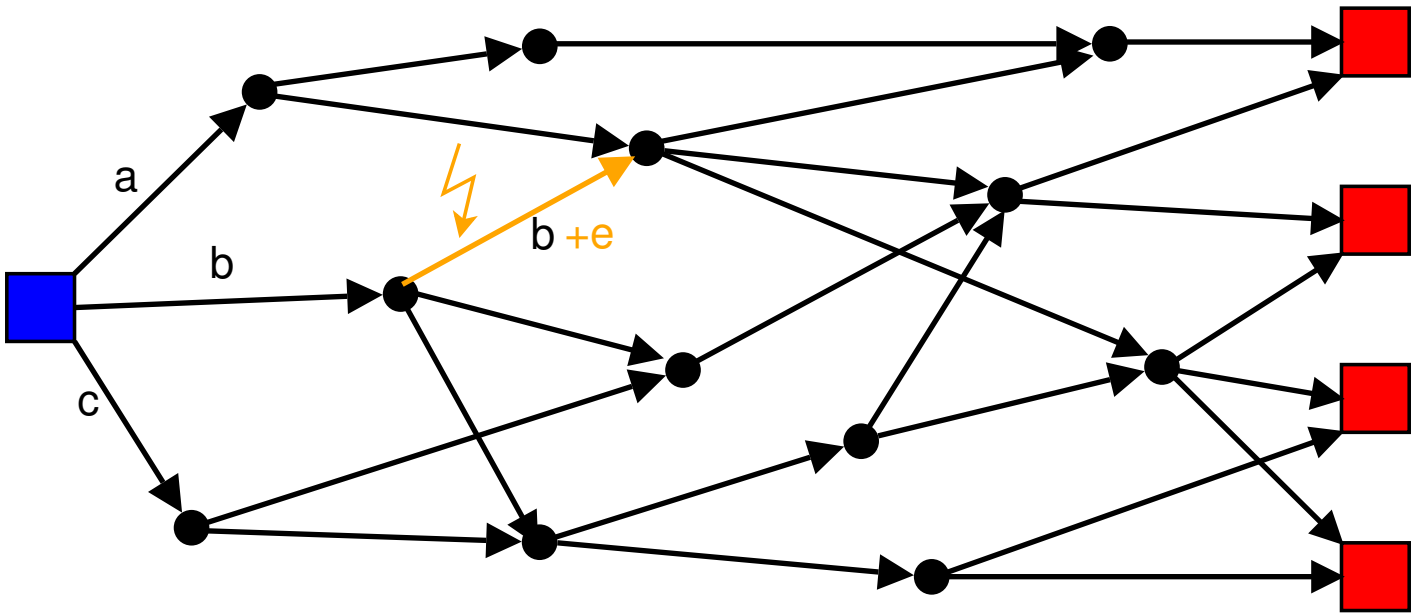
“Codieren ist besser als Weiterleiten”

Das Schmetterling-Netzwerk mit **linearem Codieren**:

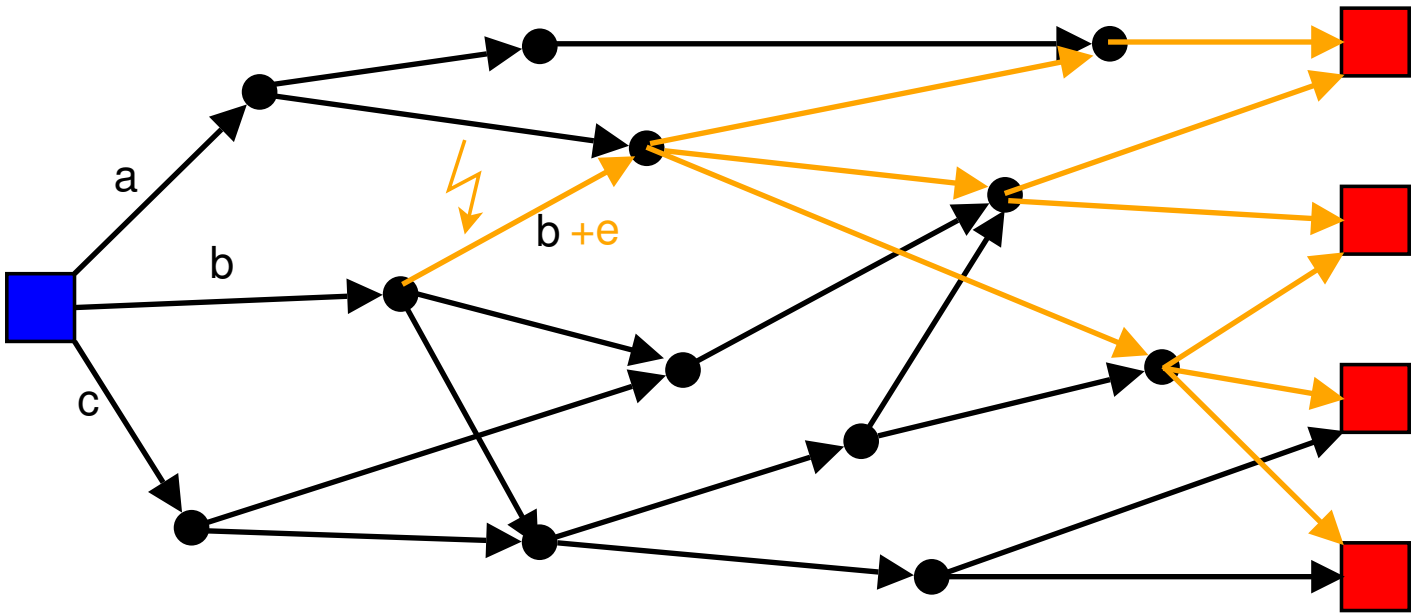


Beide Empfänger können a, b rekonstruieren.
$\Longrightarrow$ Nur 1 Zeiteinheit benötigt, um a, b an beide Empfänger zu senden.

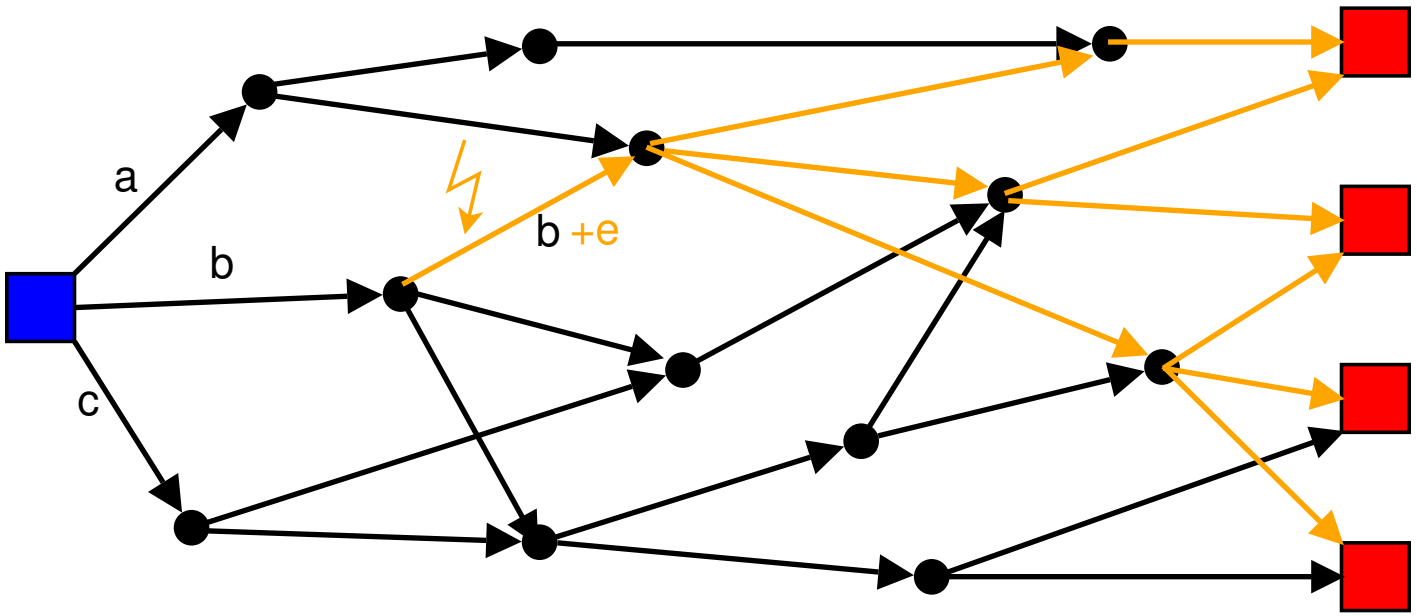
Problem: Mit linearem Codieren pflanzen sich Fehler im Netzwerk fort!



Problem: Mit linearem Codieren pflanzen sich Fehler im Netzwerk fort!



Problem: Mit linearem Codieren pflanzen sich Fehler im Netzwerk fort!



$\Longrightarrow$ Eine andere Metrik wird zum Dekodieren benötigt.

Definition

Ein *Rang-Metrik-Code* ist definiert als eine Teilmenge von $\mathbb{F}_q^{m\times n}$. Die *Rang-Metrik* ist definiert als

$$d_R(A,B) := \text{Rang}(A-B), \quad A,B \in \mathbb{F}_q^{m\times n}.$$

Definition

Ein *Rang-Metrik-Code* ist definiert als eine Teilmenge von $\mathbb{F}_q^{m\times n}$. Die *Rang-Metrik* ist definiert als

$$d_R(A,B) := \text{Rang}(A-B), \quad A,B \in \mathbb{F}_q^{m\times n}.$$

Die *Minimal-Rangdistanz* eines Codes $C \subseteq \mathbb{F}_q^{m\times n}$ ist definiert als

$$d_{\min,R}(C) := \min\{d_R(A,B) \mid A,B \in C, A \neq B\}.$$

Definition

Ein *Rang-Metrik-Code* ist definiert als eine Teilmenge von $\mathbb{F}_q^{m\times n}$. Die *Rang-Metrik* ist definiert als

$$d_R(A,B) := \text{Rang}(A-B), \quad A,B \in \mathbb{F}_q^{m\times n}.$$

Die *Minimal-Rangdistanz* eines Codes $C \subseteq \mathbb{F}_q^{m\times n}$ ist definiert als

$$d_{\min,R}(C) := \min\{d_R(A,B) \mid A,B \in C, A \neq B\}.$$

Codewörter sind jetzt Matrizen, wobei jeder Zeilen-Vektor entlang einer ausgehenden Kante vom Sender geschickt wird.

Kanal-Modell (kohärentes Network Coding)

Sei $U \in \mathbb{F}_q^{m \times n}$ ein Codewort, welches über den Kanal geschickt wurde. Ein empfangenes Wort ist von der Form

$$R = AU + E$$

wobei $A \in \mathbb{F}_q^{m \times m}$ die Linearkombinationen der inneren Knoten (von Sender und Empfänger gekannt) repräsentiert, und $E \in \mathbb{F}_q^{m \times n}$ die Fehlermatrix ist.

Kanal-Modell (kohärentes Network Coding)

Sei $U \in \mathbb{F}_q^{m \times n}$ ein Codewort, welches über den Kanal geschickt wurde. Ein empfangenes Wort ist von der Form

$$R = AU + E$$

wobei $A \in \mathbb{F}_q^{m \times m}$ die Linearkombinationen der inneren Knoten (von Sender und Empfänger gekannt) repräsentiert, und $E \in \mathbb{F}_q^{m \times n}$ die Fehlermatrix ist.

Der Empfänger dekodiert zum nächsten Codewort zu $A^{-1}R$ bzgl. der Rang-Metrik.

Verbindung zu Block-Codes:

- Über $\mathbb{F}_q^m \cong \mathbb{F}_{q^m}$ können wir Matrizen in $\mathbb{F}_q^{m\times n}$ als Vektoren in $\mathbb{F}_{q^m}^n$ darstellen.
- So kann jeder Block-Code $C \subseteq \mathbb{F}_{q^m}^n$ einen Rang-Metrik-Code erzeugen.
- So können *lineare* Rang-Metrik-Codes definiert werden (erzeugt durch lineare Block-Codes).

Beispiel

Sei $\mathbb{F}_{2^2} \cong \mathbb{F}_2[\alpha]$ mit $\alpha^2 + \alpha + 1 = 0$. Der lineare Block-Code mit Generatormatrix $G = (1\ \alpha\ 1)$ ist

$$C = \{(0\ 0\ 0), (1\ \alpha\ 1), (\alpha\ \alpha+1\ \alpha), (\alpha+1\ 1\ \alpha+1)\} \subseteq \mathbb{F}_{2^2}^3.$$

Als Rang-Metrik-Code $C^* \subseteq \mathbb{F}_2^{2\times 3}$ erhalten wir

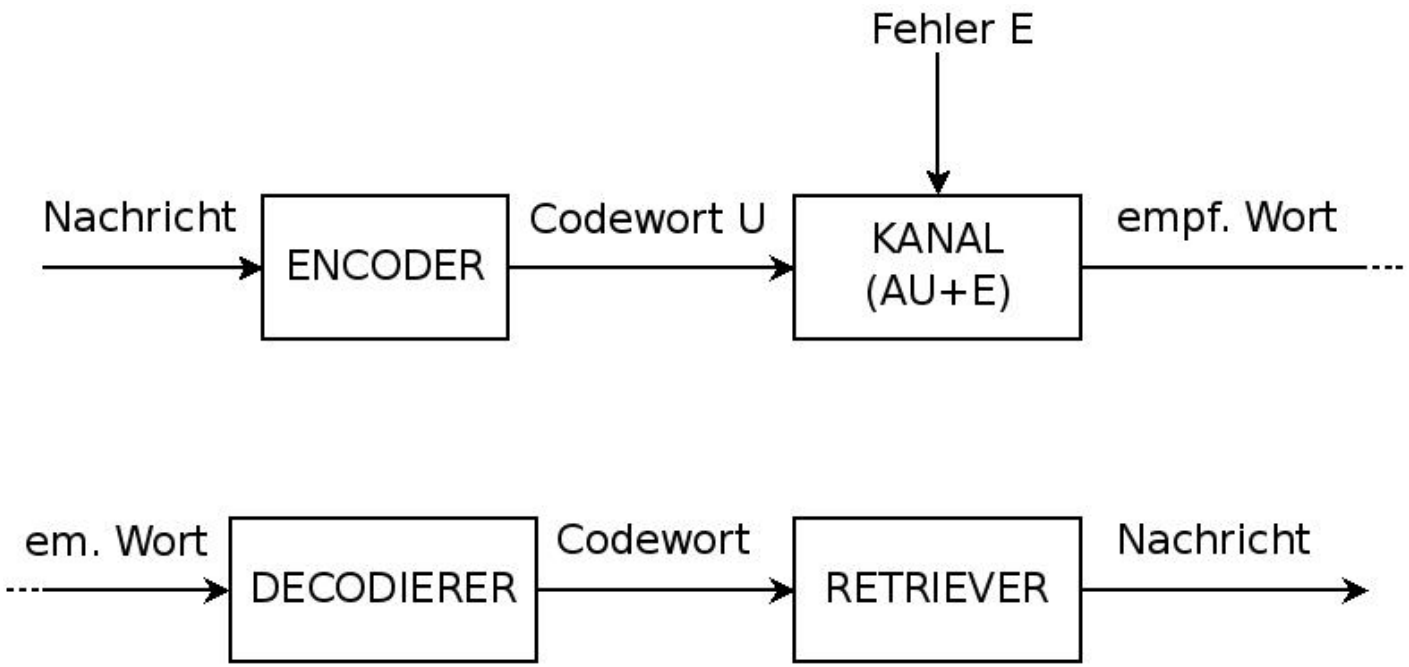
$$C^* = \left\{ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \right\}$$

mit Rang-Minimaldistanz 2.

Hauptforschungsziele (kohärentes Network Coding):

- Für einen gegebenen Raum $\mathbb{F}_q^{m\times n}$ und Minimaldistanz des Codes, finde die beste Packung bzgl. der Rang-Metrik.
  $\Longrightarrow$ beste mögliche Übertragungsrate bei gleicher Fehlerkorrektur
- Finde effiziente Decodieralgorithmen (möglicherweise zusammen mit Code-Konstruktionen).
  $\Longrightarrow$ schnellere Kommunikation

Überblick Network Coding-Prozess:



Systeme:

	Input	**Output**
Encoder	Nachricht	Codewort
Kanal	Codeword	empfangenes Wort
Decodierer	empfangenes Wort	Codewort
Retriever	Codewort	Nachricht

- Durch die Beziehung zu Block-Codes, können für Encoder und Retriever wieder linear Abbildungen verwendet werden.
- Viele Decodieralgorithmen sind ebenfalls analog zu denen in der Hamming-Metrik.
- Fehlermatrix (im Kanal) folgt Wahrscheinlichkeits-verteilung.

Decodierer + Retriever:

$$\begin{aligned}\text{Nachricht} &= \text{argmin}\{d_R(mG, \varphi_A^{-1}(r)) \mid m \in \mathbb{F}_{q^m}^k\} \\ &= \text{argmin}\{||mG - \varphi_A^{-1}(r)||_R \mid m \in \mathbb{F}_{q^m}^k\} \\ &= \text{argmin}\{||\varphi_A(mG) - r||_R \mid m \in \mathbb{F}_{q^m}^k\}\end{aligned}$$

$G \in \mathbb{F}_{q^m}^{k \times n}$ Generatormatrix
$r \in \mathbb{F}_{q^m}^n$ empfangenes Wort
$\varphi_A : \mathbb{F}_{q^m}^n \to \mathbb{F}_{q^m}^n$ repräsentiert Linearkombinationen im Netzwerk (entspricht $A : \mathbb{F}_q^{m \times n} \to \mathbb{F}_q^{m \times n}$)

Resultate aus der klassischen Codierungstheorie mit normalen Polynomen werden für die Rang-Metrik-Codes in Ring der linearisierten Polynome übertragen .

Definition

Ein *linearisiertes Polynom* in $\mathbb{F}_{q^m}[x]$ ist von der Form

$$f(x) = \sum_{i=0}^{d} f_i x^{q^i}.$$

Resultate aus der klassischen Codierungstheorie mit normalen Polynomen werden für die Rang-Metrik-Codes in Ring der linearisierten Polynome übertragen .

Definition

Ein *linearisiertes Polynom* in $\mathbb{F}_{q^m}[x]$ ist von der Form

$$f(x) = \sum_{i=0}^{d} f_i x^{q^i}.$$

Theorem

Die linearisierten Polynome in $\mathbb{F}_{q^m}[x]$ bilden einen Ring mit der üblichen Polynomaddition und Komposition als zweiter Operation.

Theorem

- *Linearisierte Polynome in* $\mathbb{F}_{q^m}[x]$ *sind* $\mathbb{F}_q$*-lineare Abbildungen.*
- *Die Nullstellen eines linearisierten Polynoms bilden einen* $\mathbb{F}_q$*-Vektorraum.*
- *Zu jeden* $\mathbb{F}_q$*-Vektorraum* $V \subseteq \mathbb{F}_{q^m}$ *ist* $\prod_{\beta \in V}(x - \beta)$ *ein linearisiertes Polynom.*

Theorem

- *Linearisierte Polynome in* $\mathbb{F}_{q^m}[x]$ *sind* $\mathbb{F}_q$*-lineare Abbildungen.*
- *Die Nullstellen eines linearisierten Polynoms bilden einen* $\mathbb{F}_q$*-Vektorraum.*
- *Zu jeden* $\mathbb{F}_q$*-Vektorraum* $V \subseteq \mathbb{F}_{q^m}$ *ist* $\prod_{\beta \in V}(x - \beta)$ *ein linearisiertes Polynom.*

Beispiel in $\mathbb{F}_{3^2} \cong \mathbb{F}_3[\alpha]$

- Das linearisierte Polynom $x^3 - \alpha x$ hat Nullstellen $0, \alpha^2, 2\alpha^2$.
- Komposition: $(x^3 - \alpha x) \circ (\alpha x^3) = \alpha^3 x^9 - \alpha^2 x^3$

$$(\alpha x^3) \circ (x^3 - \alpha x) = \alpha x^9 - \alpha^2 x^3$$

Nicht kommutativ!!

Theorem

- *Linearisierte Polynome in* $\mathbb{F}_{q^m}[x]$ *sind* $\mathbb{F}_q$*-lineare Abbildungen.*
- *Die Nullstellen eines linearisierten Polynoms bilden einen* $\mathbb{F}_q$*-Vektorraum.*
- *Zu jeden* $\mathbb{F}_q$*-Vektorraum* $V \subseteq \mathbb{F}_{q^m}$ *ist* $\prod_{\beta \in V}(x - \beta)$ *ein linearisiertes Polynom.*

Beispiel in $\mathbb{F}_{3^2} \cong \mathbb{F}_3[\alpha]$

- Das linearisierte Polynom $x^3 - \alpha x$ hat Nullstellen $0, \alpha^2, 2\alpha^2$.
- Komposition: $(x^3 - \alpha x) \circ (\alpha x^3) = \alpha^3 x^9 - \alpha^2 x^3$

$$(\alpha x^3) \circ (x^3 - \alpha x) = \alpha x^9 - \alpha^2 x^3$$

Nicht kommutativ!!

$\Longrightarrow$ Inverse Operation nennt man *symbolische Divison.*

Auf zum **linearisierten Berlekamp-Massey-Algorithmus**!

Lemma

Wenn $E \in \mathbb{F}_q^{m \times n}$ *Rang* t *hat, gibt es* $A \in \mathbb{F}_q^{m \times t}, B \in \mathbb{F}_q^{t \times n}$ *mit*

$$E = AB.$$

(Eindeutig bis auf GL_t*-Operation in Mitte.)*

Lemma

Wenn $E \in \mathbb{F}_q^{m \times n}$ *Rang* t *hat, gibt es* $A \in \mathbb{F}_q^{m \times t}, B \in \mathbb{F}_q^{t \times n}$ *mit*

$$E = AB.$$

(Eindeutig bis auf GL_t*-Operation in Mitte.)*

- Isomorph: $e = aB$ mit $e \in \mathbb{F}_{q^m}^n, a \in \mathbb{F}_{q^m}^t$
- Alle Elemente in $\langle a_1, \ldots, a_t \rangle_{\mathbb{F}_q}$ sind Nullstellen eines linearisierten Polynoms $\Lambda(x)$ vom q-Grad t.
- $\Lambda(x) = \sum_{i=0}^{t} f_i x^{q^i}$ nennt man *Error-Span-Polynom.*

Adaption von Berlekamp-Massey-Algorithmus:

- Repräsentiere den *Gabidulin-Code* $C \subseteq \mathbb{F}_q^{m\times n}$ als Block-Code in $\mathbb{F}_{q^m}^n$ mit Kontrollmatrix

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^n \\ 1 & \alpha^q & \alpha^{2q} & \dots & \alpha^{nq} \\ \vdots & & & & \vdots \\ 1 & \alpha^{q^{n-k-1}} & \alpha^{2q^{n-k-1}} & \dots & \alpha^{nq^{n-k-1}} \end{pmatrix}.$$

- Repräsentiere die empfangene Matrix $R = AU + E \in \mathbb{F}_q^{m\times n}$ als $r = \varphi_A(c) + e \in \mathbb{F}_{q^m}^n$, wobei $\mathrm{Rang}(E) = t$.
- Berechne die Syndrome $(s_1, s_2, \dots, s_{n-k}) = rH^T = eH^T$.

Adaption von Berlekamp-Massey-Algorithmus:

- Finde das linearisierte Polynom $\Lambda(x) = \sum \Lambda_i x^{q^i} \in \mathbb{F}_{q^m}[x]$ vom kleinsten Grad, s.d.

$$\Lambda_0 s_{i+t} + \Lambda_1 s_{i+t-1}^{q} + \Lambda_2 s_{i+t-2}^{q^2} + \cdots + \Lambda_t s_i^{q^t} = 0$$

für $i = 0, \ldots, n-k-t$.

Adaption von Berlekamp-Massey-Algorithmus:

- Finde das linearisierte Polynom $\Lambda(x) = \sum \Lambda_i x^{q^i} \in \mathbb{F}_{q^m}[x]$ vom kleinsten Grad, s.d.

$$\Lambda_0 s_{i+t} + \Lambda_1 s_{i+t-1}^{q} + \Lambda_2 s_{i+t-2}^{q^2} + \cdots + \Lambda_t s_i^{q^t} = 0$$

  für $i = 0, \ldots, n-k-t$.
- Bestimmte eine Basis $\{a_1, \ldots, a_t\}$ des Nullstellen-Raums von $\Lambda(x)$.
- Löse das über $\mathbb{F}_q$ expandierte Gleichungssystem

$$\underbrace{(a_1, \ldots, a_t)B}_{E} H^T = (s_1, \ldots, s_{n-k}).$$

- Decodiere zu $U = A^{-1}(R - E)$.

Beispiel

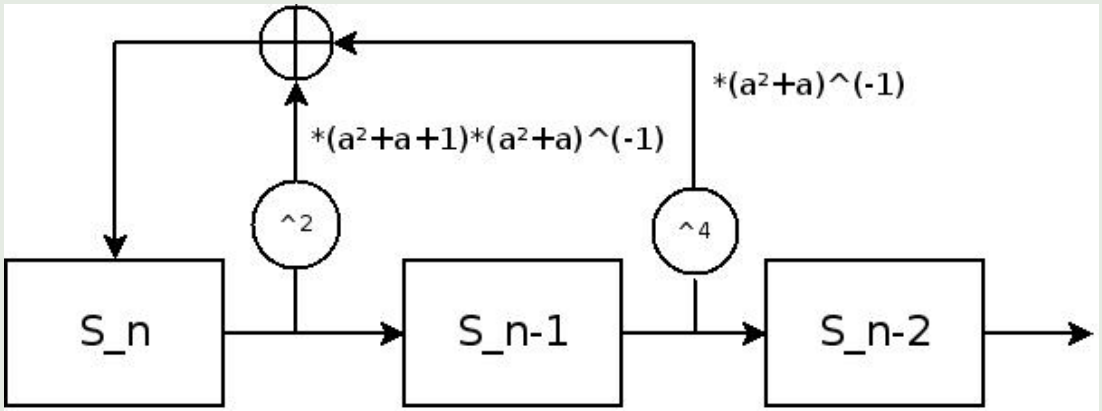
Wir betrachten $\mathbb{F}_{2^5} = \mathbb{F}_2[\alpha]$ und haben die Syndromfolge

$$(s_1, s_2, s_3, s_4) = (\alpha^2 + 1, \alpha^3 + 1, \alpha^5 + 1, \alpha^9 + 1).$$

Der Berlekamp-Massey-Typ-Algorithmus ermittelt

$$\Lambda(x) = x^4 + (\alpha^2 + \alpha + 1)x^2 + (\alpha^2 + \alpha)x.$$

Dies entspricht keinem linear rückgekoppeltem Schieberegister, aber dem q-analogen eines solchen:



Beispiel fortgesetzt

Die Nullstellen von

$$\Lambda(x) = x^4 + (\alpha^2 + \alpha + 1)x^2 + (\alpha^2 + \alpha)x$$

sind $\{0, 1, \alpha, \alpha + 1\} = \langle 1, \alpha \rangle_{\mathbb{F}_2}$. Dann können wir

$$(1, \alpha)BH^T = (\alpha^2 + 1, \alpha^3 + 1, \alpha^5 + 1, \alpha^9 + 1)$$

über $\mathbb{F}_2$ lösen, $E = (1, \alpha)B$ rekonstruieren und somit decodieren.

Zweites Beispiel: **Welch-Berlekamp-Algorithmus**

Welch-Berlekamp-Algorithmus (linearisiert):

- Betrachte einen Gabidulin-Code $C \subseteq \mathbb{F}_{q^m}^n$ mit Generatormatrix

$$G = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^{q^2} & g_2^{q^2} & \cdots & g_n^{q^2} \\ \vdots & & & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}$$

  und das empfangene Wort $r = (r_1, r_2, \ldots, r_n) \in \mathbb{F}_{q^m}^n$.
- Erstelle das linearisierte *Lagrange-Polynom* $\Lambda(x)$, s.d.

$$\Lambda(g_i) = r_i$$

  und das linearisierte *Annullatorpolynom* $\Pi(x)$, s.d.

$$\Pi(g_i) = 0.$$

Welch-Berlekamp-Algorithmus (linearisiert):

- Erstelle Modul $\mathfrak{M}$ (über dem Ring der linearisierten Polynome) mit der Zeilen-Basis

$$\begin{bmatrix} \Pi(x) & 0 \\ -\Lambda(x) & x \end{bmatrix}.$$

- Finde minimale (reduzierte) Basis $(b_1\ b_2), (b'_1\ b'_2)$ für $\mathfrak{M}$ (bzgl. des $(0, k-1)$-gewichteten q-Grads).
- Dann ist $b'_1 \circ^{-1} b'_2$ die dekodierte Nachricht.

Welch-Berlekamp-Algorithmus (linearisiert):

- Erstelle Modul $\mathfrak{M}$ (über dem Ring der linearisierten Polynome) mit der Zeilen-Basis

$$\begin{bmatrix} \Pi(x) & 0 \\ -\Lambda(x) & x \end{bmatrix}.$$

- Finde minimale (reduzierte) Basis $(b_1\ b_2), (b'_1\ b'_2)$ für $\mathfrak{M}$ (bzgl. des $(0, k-1)$-gewichteten q-Grads).
- Dann ist $b'_1 \circ^{-1} b'_2$ die dekodierte Nachricht.

$\Longrightarrow$ Modellierung in Moduln von minimalem Grad
$=$ *minimale Zustandsdarstellung*

Welch-Berlekamp-Algorithmus (linearisiert):

- Erstelle Modul $\mathfrak{M}$ (über dem Ring der linearisierten Polynome) mit der Zeilen-Basis

$$\begin{bmatrix} \Pi(x) & 0 \\ -\Lambda(x) & x \end{bmatrix}.$$

- Finde minimale (reduzierte) Basis $(b_1\ b_2), (b'_1\ b'_2)$ für $\mathfrak{M}$ (bzgl. des $(0, k-1)$-gewichteten q-Grads).
- Dann ist $b'_1 \circ^{-1} b'_2$ die dekodierte Nachricht.

$\Longrightarrow$ Modellierung in Moduln von minimalem Grad
$=$ *minimale Zustandsdarstellung*

$\Longrightarrow$ kann mit linearisiertem Euklid oder Gröbnerbasen berechnet werden

Zusammenfassung

- Einführung klassische Codierungstheorie (Hamming-Metrik)
- Einführung kohärentes Network Coding (Rang-Metrik)
- Beispiele von verschiedenen Systemen:
  - Encoder, Kanal, Decodierer, Retriever
  - Berlekamp-Massey-Decodieralgorithmus (klassisch und linearisiert)
  - Welch-Berlekamp-Decodieralgorithmus (linearisiert)
- Häufige Technik: Minimale Zustandsdarstellung / Irreduzible Transferfunktion

Ausblick in andere Bereiche

- Belief-Propagation-Algorithmus (System mit Feedback, was stabilisiert werden soll)
- Faltungscodes (Encoder ist System mit Feedback)
- Physical Layer Coding / Gitter-Codierungstheorie (über $\mathbb{R}^n$ oder $\mathbb{C}^n$ mit Euklidischer Metrik)
- Coding über Fading Channels

Ausblick in andere Bereiche

- Belief-Propagation-Algorithmus (System mit Feedback, was stabilisiert werden soll)
- Faltungscodes (Encoder ist System mit Feedback)
- Physical Layer Coding / Gitter-Codierungstheorie (über $\mathbb{R}^n$ oder $\mathbb{C}^n$ mit Euklidischer Metrik)
- Coding über Fading Channels

Danke für die Aufmerksamkeit!
Fragen? – Kommentare?