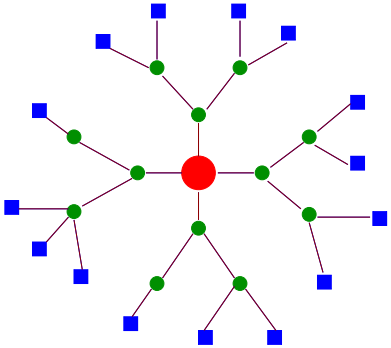
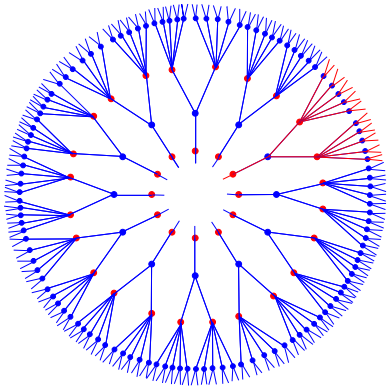
# Content Delivery und Codierungstheorie





Amin Shokrollahi

digitalfountain

# Inhalt

- Problemstellung
- Standardlösungen
- Codierungstheoretische Lösung
- LDPC-Codes und ihre Analyse
- Tornado Codes
- Ausblick

# Problemstellung

Ziel: Wollen grosse Datenmengen über ein Computernetzwerk an viele Benutzer anliefern.

- Neueste Version des IE-Explorers
- Neueste Versionen von Spielen
- ...

Problem: Benutzer benötigen exakte Kopie des Originals. Daten gehen aber während der Übertragung verloren.

Wollen: Zuverlässige Übetragungslösungen

# Kostenmasse und Skalierbarkeit

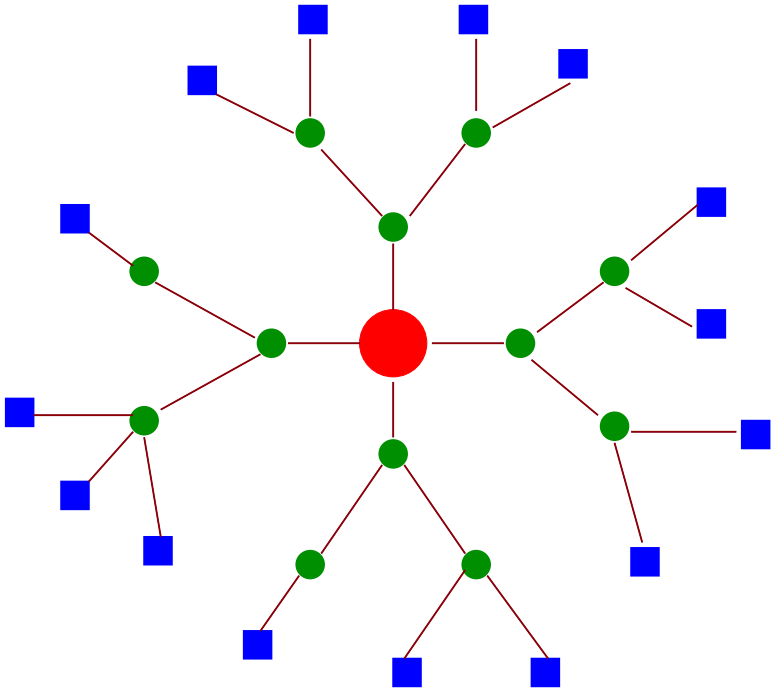
Kostenmasse:

- Anzahl der Server
- Bandbreite aus dem Serverzentrum

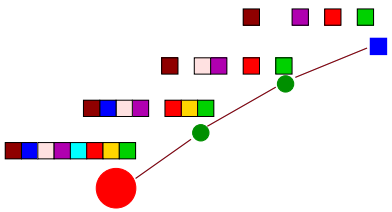
Eine Übertragungslösung heisst *skalierbar*, wenn die Kosten der Übetragung nicht mit der Anzahl der Benutzer wachsen. (Server-skalierbar, Bandbreiten-skalierbar.)

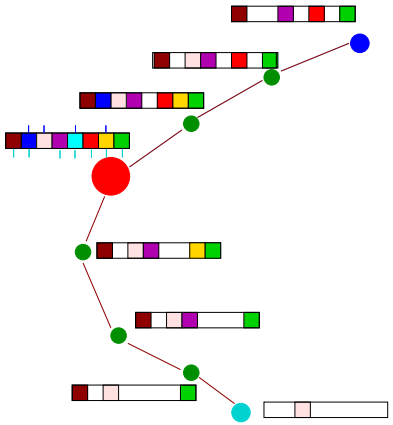
Sind interessiert an skalierbare Lösungen, die gleichzeitig zuverlässig sind.

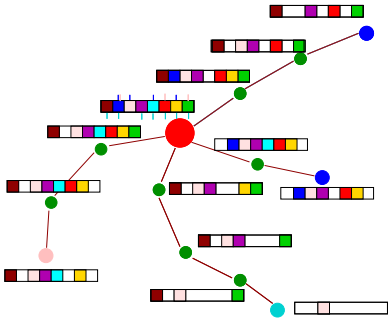
# TCP/IP



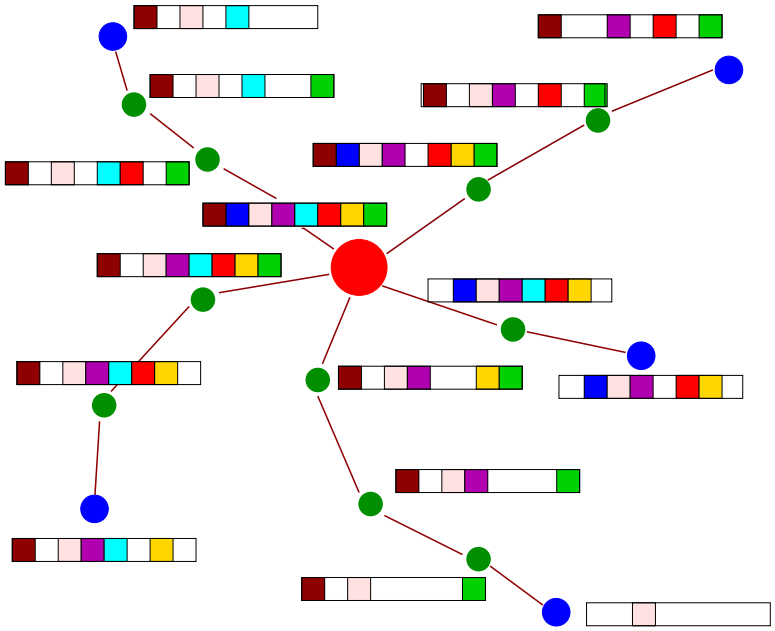
# TCP/IP





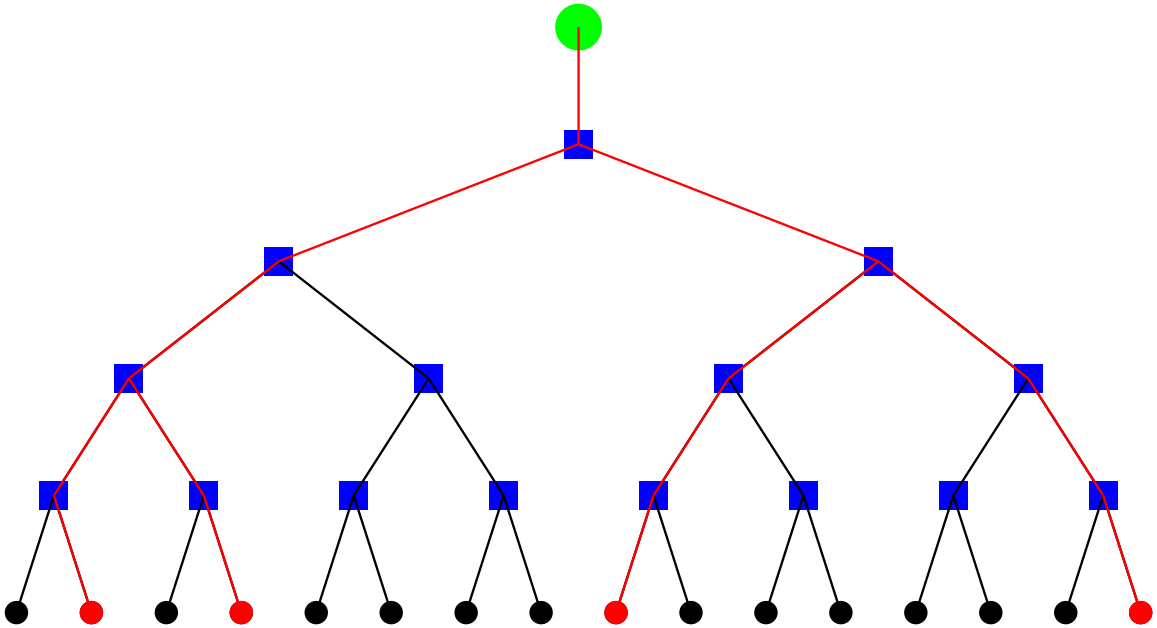


# TCP/IP



TCP/IP ist zuverlässig, aber weder Server- noch Bandbreiten-skalierbar.

# Multicast



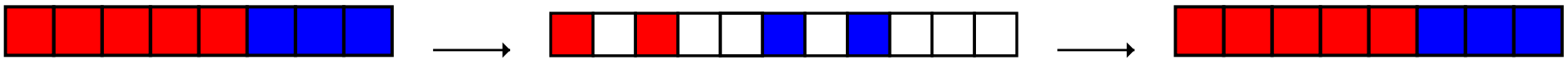
Multicast is Server- und Bandbreiten-skalierbar, aber nicht zuverlässig.

# Lösung: Codes

Wollen die Vorteile von Multicast und TCP/IP, aber nicht deren Nachteile.

Codiere die Originaldaten und schicke die codierte Version via Multicast.

Rekonstruktion ist möglich, falls nicht zu viele Pakete verloren gehen.

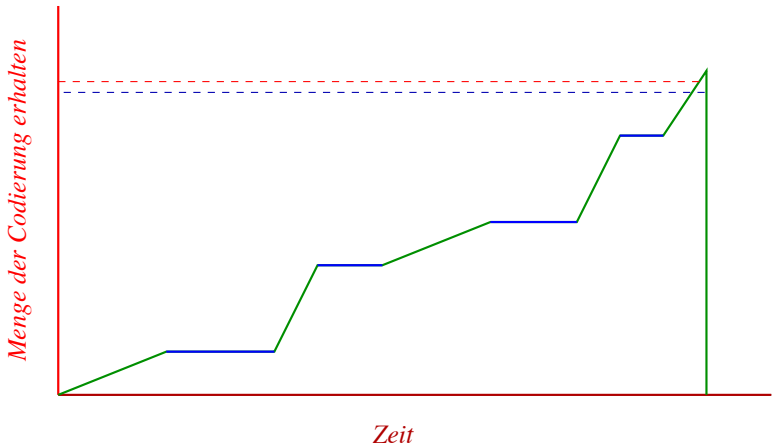


Zuverlässigkeit $\rightarrow$ Codierung.

Skalierbarkeit $\rightarrow$ Multicast.

# Lösung

Carousel-Technik



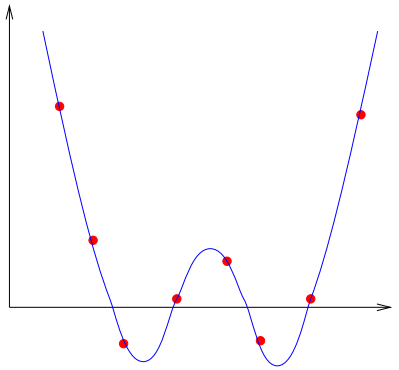
# Lineare Codes

Ein linearer Code der Blocklänge $n$ und Dimension $k$ über dem endlichen Körper $\mathbb{F}_q$ ist ein $k$-dimensionaler Teilraum von $\mathbb{F}_q^n$.

Beispiel: Reed-Solomon Codes über dem Körper $\mathbb{F}_q$.

- Wähle Werte $x_1, x_2, \ldots, x_n$ in $\mathbb{F}_q$, paarweise verschieden.
- RS-Code der Dimension $k$ ist Bild der Abbildung

$$\mathbb{F}_q[x]_{<k} \to \mathbb{F}_q^n, \qquad f \mapsto (f(x_1), f(x_2), \ldots, f(x_n)).$$



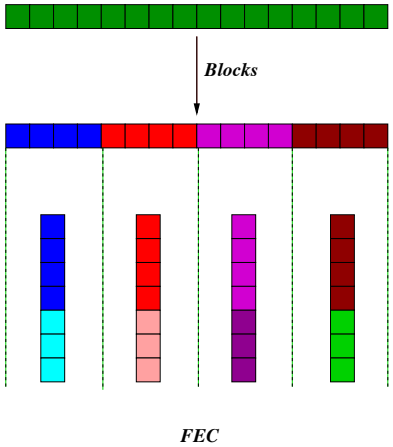
# Effizienz, Zuverlässigkeit und traditonelle Codes

Reed-Solomon Codes:

- Interpretiere Originaldaten als Koeffizienten eines Polynoms.
- Werte das Polynom an genügend vielen Punkten des Grundkörpers aus.
- Decodiere durch Interpolation. (Original kann aus jeder Menge von $k$ empfangenen Koordinaten rekonstruiert werden.)

Reed-Solomon Codes sind nicht effizient für grosse Datenmengen. $O(n^2)$.

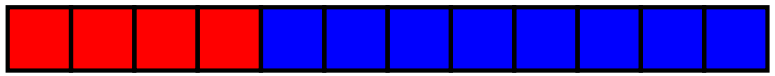
Müssen die Originaldaten in Blöcke unterteilen. Verlieren Zuverlässigkeit.



# Idealfall

Brauchen *binäre* Codes, die sehr schnell ($O(n)$) codier- und decodierbar sind.

Können solche Codes konstruieren. Sie haben aber notwendigerweise ein Overhead: Das Original kann aus *meisten* Mengen von $k \cdot (1 + \varepsilon)$ Koordinaten berechnet werden.

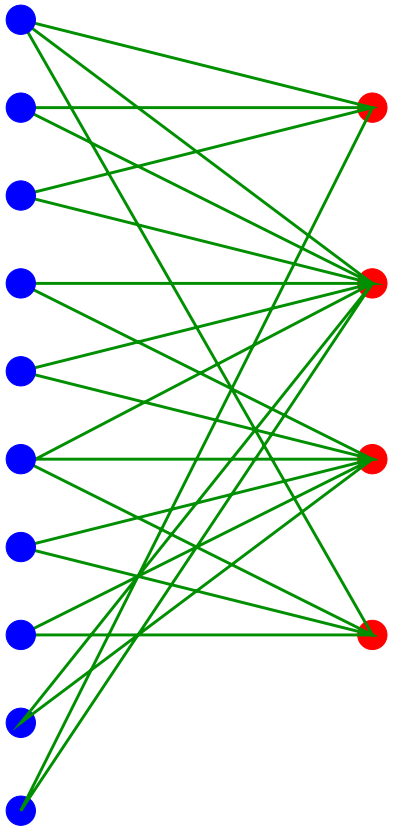


Perfekt

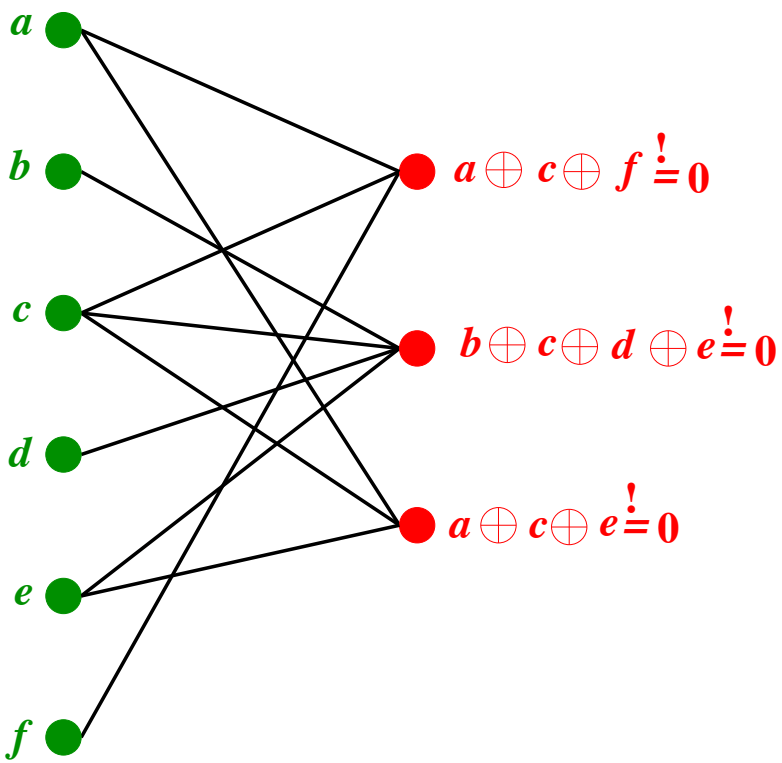
$k \cdot (1 + \varepsilon)$

# LDPC-Codes

LDPC-Codes werden mittels dünner bipartiter Graphen konstruiert.



# Konstruktion




Jeder binäre lineare Code hat eine graphische Darstellung.

Nicht jeder Code kann durch einen dünnen Graphen dargestellt werden.

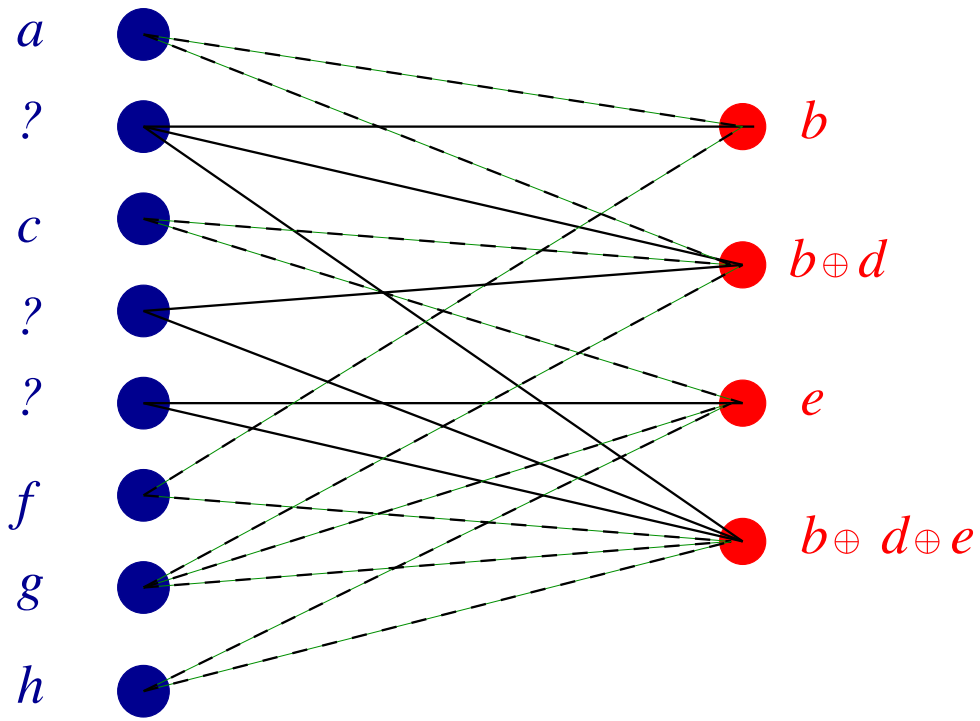
# Algorithmische Fragen

- Codierzeit
  - Ist linear für die duale Konstruktion
  - Ist quadratisch (nach Vorbereitung) für die Gallager Konstruktion. Mehr dazu später!
- Decodierung?

# Decodieren

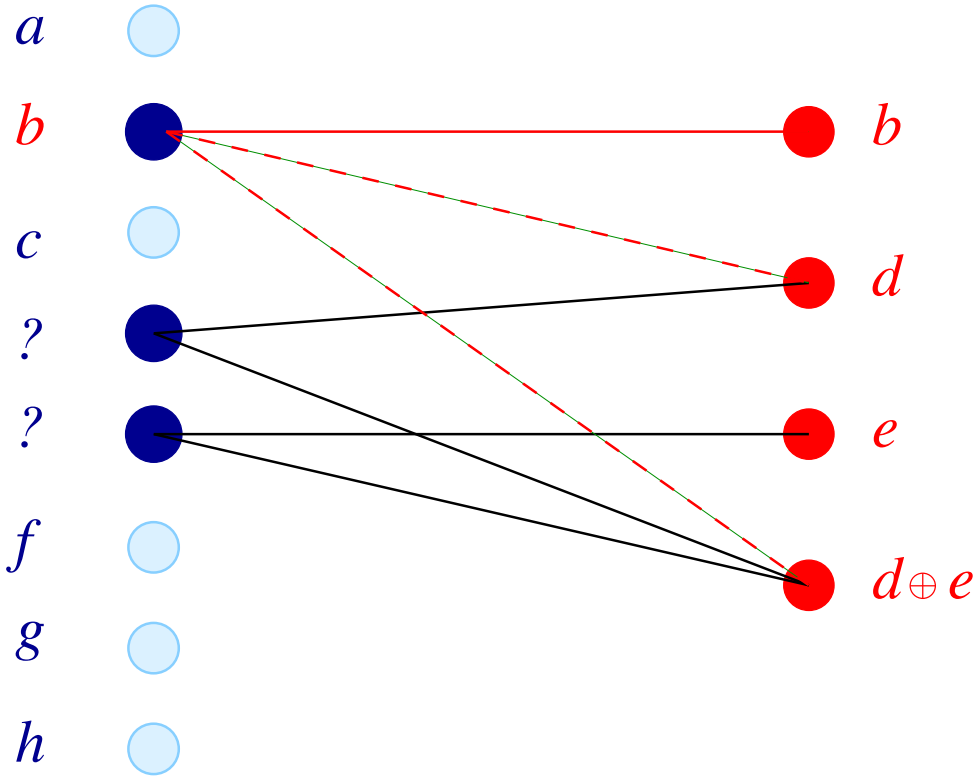
Luby-Mitzenmacher-Shokrollahi-Spielman, 1997:

Phase 1: Direkte Berechnung

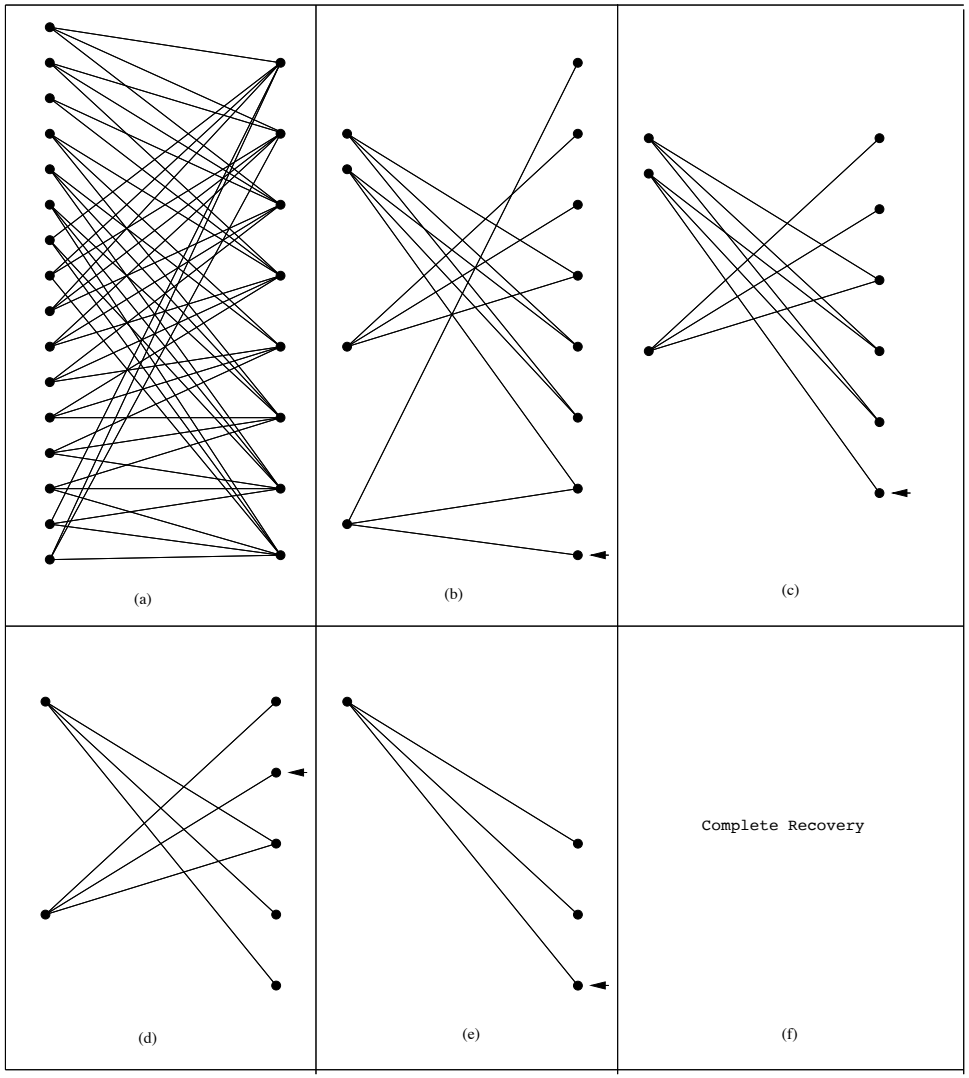


# Decodieren

Phase 2: Substitution



# Beispiel



# Das (inverse) Problem

Haben: schnellen Decodieralgorithmus.

Wollen: Codes entwerfen die viele Fehler mittels dieses Algorithmus korrigieren können.

# Experimente

Wähle reguläre Graphen.

Ein $(d, k)$-regulärer Graph kann höchstens einen $d/k$-Anteil von Auslöschungen korrigieren.

Wähle zufälligen $(d, k)$-Graphen.

$p_0$ := Maximaler Anteil von korrigierbaren Auslöschungen.

| $d$ | $k$ | $d/k$ | $p_0$ |
|---|---|---|---|
| 3 | 6 | 0.5 | 0.429 |
| 4 | 8 | 0.5 | 0.383 |
| 5 | 10 | 0.5 | 0.341 |
| 3 | 9 | 0.33 | 0.282 |
| 4 | 12 | 0.33 | 0.2572 |

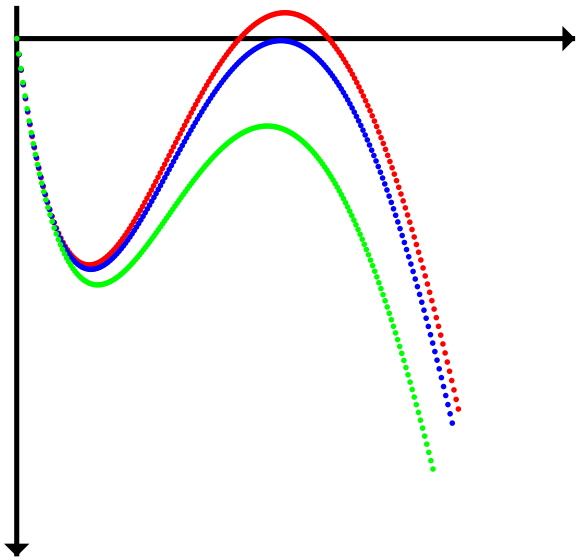
Wo kommen die Zahlen her?

# Theorem

(Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann 1997) Ein zufällig gewählter $(d, k)$-Graph kann einen $p_0$-Anteil von Auslöschungen mit hoher Wahrscheinlichkeit korrigieren dann und nur dann, wenn

$$p_0 \cdot (1 - (1 - x)^{k-1})^{d-1} < x \qquad \text{für } x \in (0, p_0).$$

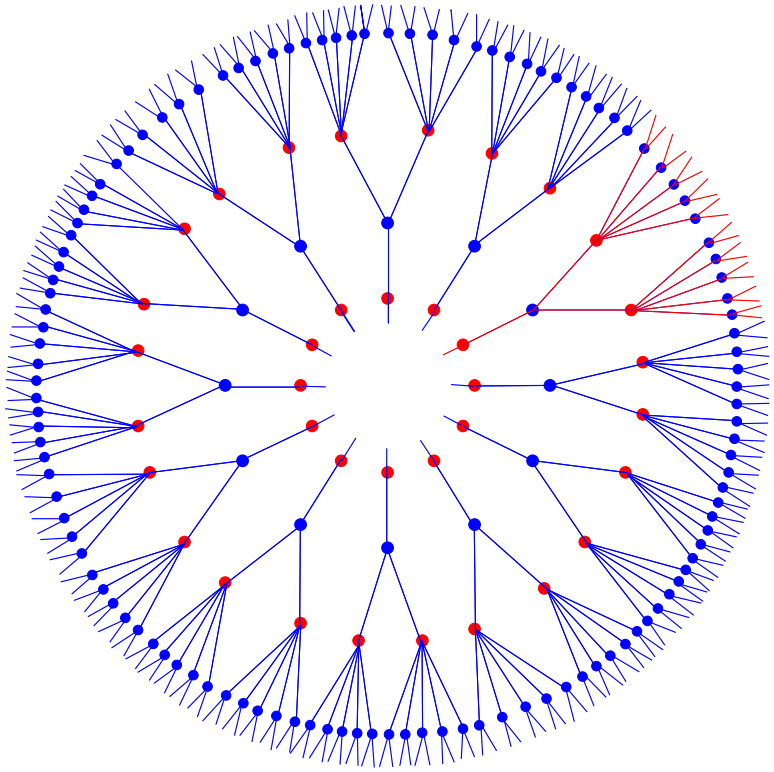
$d = 3, k = 6$: $\quad f(x) = p_0(1 - (1 - x)^5)^2 - x$



$p_0 = 0.435$
$p_0 = 0.429$
$p_0 = 0.41$

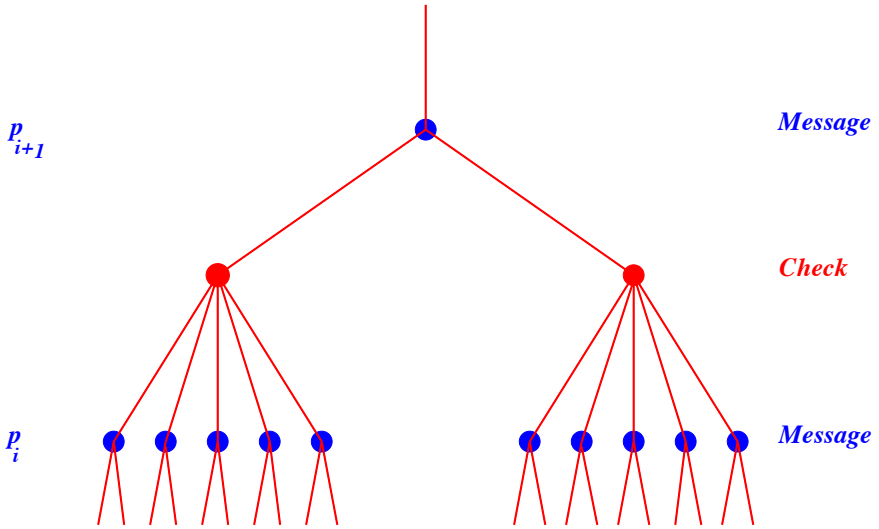
# Analyse: $(3, 6)$-Graphen

Expandiere Nachbarschaft von Nachrichtenknoten.



# Analyse: $(3, 6)$-Graphen

$p_i$ = Wahrscheinlichkeit, dass Nachrichtknoten nach der $i$-ten Iteration nicht korrigiert ist.




$$p_{i+1} = p_0(1 - (1 - p_i)^5)^2 < p_i.$$

# Analyse: $(3, 6)$-Graphen

Rigoroses Argument:

- Nachbarschaft ist Baum: Mit hoher Wahrscheinlichkeit. Standardargument.

- Obiges Argument funktioniert für erwarteten Anteil von Auslöschungen in der $i$-ten Iteration.

  Eigentlicher Wert ist konzentriert um den Erwartungswert $p_\ell$: Edge exposure martingale, Azumas Ungleichung.

# Der allgemeine Fall

$\lambda_i$ und $\rho_i$ Anteil der Kanten vom Grad $i$ auf der linken und der rechten Seite des Graphen.

$\lambda(x) := \sum_i \lambda_i x^{i-1}$, $\rho(x) := \sum_i \rho_i x^{i-1}$.

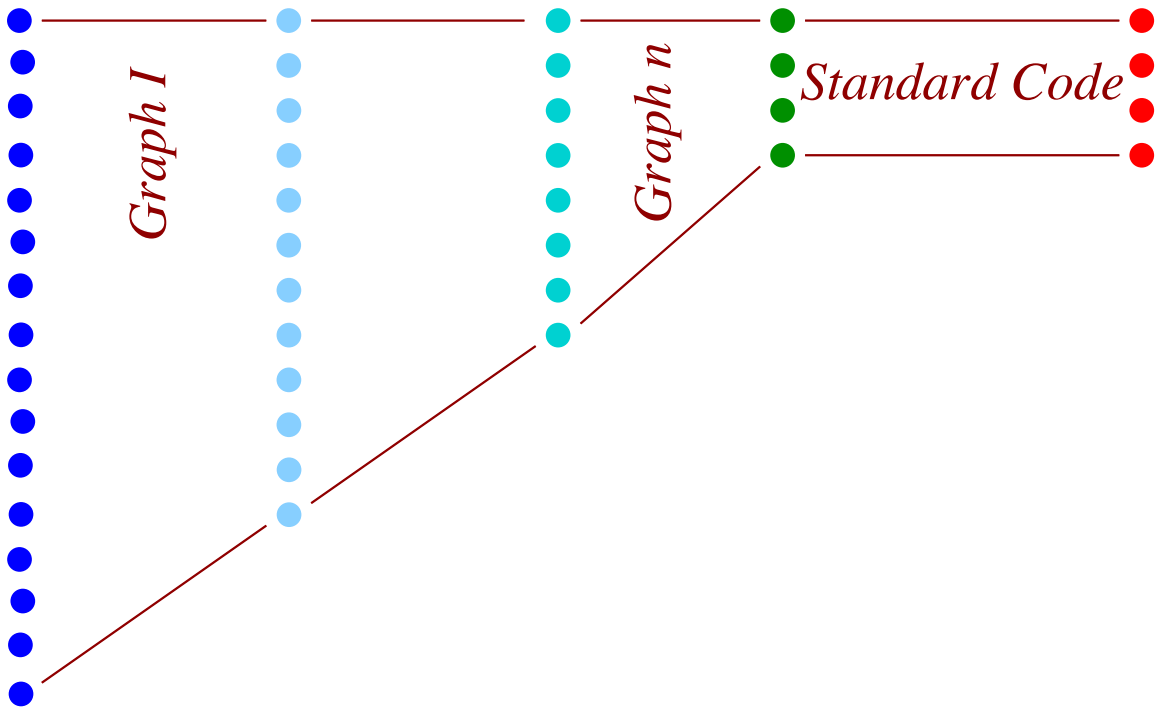
Bedingung für erfolgreiches Decodieren bei Auslöschungswahrscheinlichkeit $p_0$:

$$p_0 \lambda\left(1 - \rho(1-x)\right) < x$$

für alle $x \in (0, p_0)$.

# Codierung

Spielman, 1995: Trivial für duale Konstruktion, aber Kasakde von Graphen nötig.



# Codierung: Gallager Konstruktion

Trick: Verwende Auslöschungskorrektur.

Richaradson-Urbanke, 1999: Unter leichten Bedingungen ist Codierung in Linearzeit möglich.

Genauer brauchen wir:

- Genug Nachrichtenknoten vom Grad 2: $\lambda_2 \rho'(1) > 1$. (Grosse Komponente eines zufälligen Graphen.)

- $\rho(1 - \lambda(1 - x)) < x$ für $x \in (0, 1)$.

$(3, 6)$-Graph kann in Zeit $(0.07n)^2$ codiert werden.

# Tornado Codes

Wollen Codes entwerfen, die asymptotisch einen optimalen Anteil von Auslöschungen korrigieren können.

Wollen $\lambda$ und $\rho$ so entwerfen, daß

$$p_0 \lambda(1 - \rho(1 - x)) < x$$

für alle $x \in (0, p_0)$, und $p_0$ beliebig nahe an

$$1 - R = \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}.$$

# Tornado Codes

Extrem irreguläre Graphen, die für jede Rate $R$ Folgen von Codes liefern, die beliebig nahe an der Kapazität des Auslöschungskanals sind.

Gradstruktur?

Wähle Entwurfsparameter $D$.

$$\lambda(x) := \frac{1}{H(D)}\left(x + \frac{x^2}{2} + \cdots + \frac{x^D}{D}\right)$$

$$\rho(x) := \exp\left(\mu(x-1)\right),$$

wobei $H(D)$ die harmonische Summe $1 + 1/2 + \cdots + 1/D$ ist und $\mu = H(D)/\left(1 - 1/(D+1)\right)$.

# Tornado Codes: Effizienz

Tornado Codes können in Linearzeit codiert und decodiert werden.

Brauchen $k \cdot (1 + \varepsilon)$ Koordinatenstellen, um zu decodieren.

Die Laufzeit des Codier- und Decodieralgorithmus ist $O(n \log(1/\varepsilon))$.

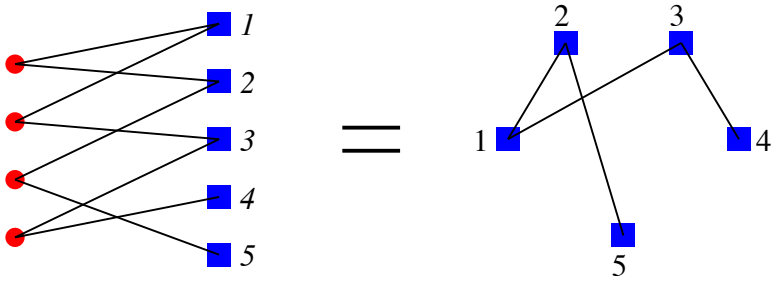
Das Trade-Off zwischen $\varepsilon$ und Laufzeit ist optimal.

# Anwendungen

- Theorie:
  - "Kapazitätsoptimale" Folgen über dem Auslöschungskanal
  - Zusammenhang mit zufälligen Graphen
  - Codes über anderen Übertragungskanäle
  - Algebraische Konstruktion von LDPC Codes (Expander)
  - Analyse endlicher Codes
  - Kryptographie
  - ...

# Zusammenhang mit zufälligen Graphen

Betrachte Codes vom Grad 2 auf der linken Seite. Können dies als Graphen auf den rechten Knoten auffassen.



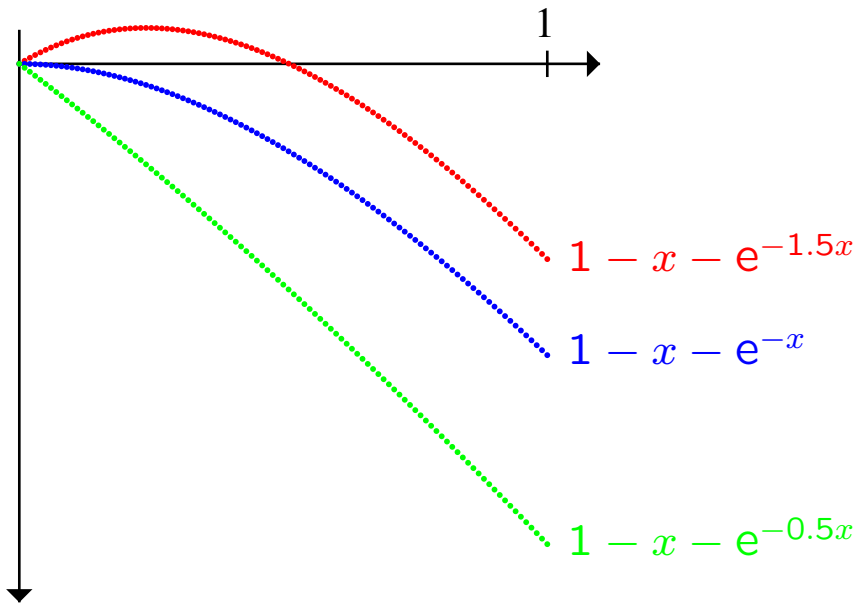

Wähle Nachbarn auf der rechten Seite zufällig (Poisson-Verteilung). Erhalten einen zufälligen Graphen auf den rechten Knoten.

$\lambda(x) = x$, $\rho(x) = \mathrm{e}^{a(x-1)}$, $a$ ist durchschnittlicher Grad.

Bedingung: $\lambda(1 - \rho(1 - x)) < x$, i.e., $1 - x - \mathrm{e}^{-ax} < 0$.

# Zusammenhang mit zufälligen Graphen

Grösste Lösung von $1 - x - \mathrm{e}^{-ax} = 0$ im Intervall $(0, 1)$ gibt Anteil der grössten Komponente. Haben grosse Komponente, wenn $a > 1$. Genau dann kann der Decodierer nicht alle Auslöschungen korrigieren.



# Zusammenhang mit zufälligen Graphen

Der Zusammenhang mit zufälligen Graphen ist kein Zufall sondern organisch.

Der Zusammenhang existiert, auch wenn der Grad der rechten Knoten nicht gleich 2 ist.

Die Tornado Folge kann aus diesem Zusammenhang durch Anwendung einer speziellen *Selbstähnlichkeitstechnik* hergeleitet werden.

Dieser Zusammenhang ergibt auch Linearzeit-Codierer für Tornado Codes.

# Praktische Anwendungen

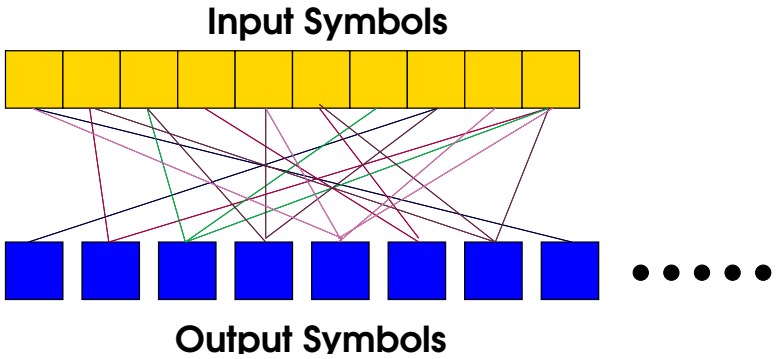
- Congestion Control
- Download von verschiedenen Servern
- Peer-to-peer Netzwerke
- Multi-Path Delivery
- Video-on-Demand
- Datenanlieferung über grosse Entfernungen
- Robustes Speichern

# Beyond Tornado

Für den Aufbau von kommerziellen Systemen sind traditionelle Codes, darunter sogar auch die Tornado Codes weniger geeignet, da der Server die Auslöschungsrate von individuellen Benutzern in Betracht ziehen muss.

Brauchen *universelle* Codes, die gleichzeitig für alle Benutzer optimal sind.

Luby (LT)-Codes:



# LT-Codes

Balls und Bins Problem ergibt: Durchschnittsgrad der Output-Knoten ist $\Omega(\log(n))$, wobei $n$ die Anzahl der Input-Symbole ist. where $n$ is number of input symbols.

LT-Codes haben Durchschnittsgrad $O(\log(n))$, und können in Zeit $O(n \log(n))$ codiert und decodiert werden.

Die Theorie von LT-Codes unterscheidet sich in einigen Punkten von der Theorie der Tornado Codes.

# Fazit

- Lineare Codes können auf exzellente Weise zum skalierbaren Übetragen von Informationen über dem Internet verwendet werden.

- Eine eigens hierfür entworfene Klasse von Codes leistet das Gewünschte.

- Die Analyse dieser Codes basiert auf Methoden der theoretischen Informatik und führt zu interessanten Forschungsfragen innerhalb der Codierungs- und der Graphentheorie.

- Die theoretischen Resultate lassen sich sofort in die Praxis umsetzen. Anwendungen gehen weit über das Problem der skalierbaren Datenübertragung hinaus.

# Kapazitätsoptimale Folgen über dem Auslöschungskanal

Klassifiziere alle Folgen $\lambda^{(n)}$ und $\rho^{(n)}$ mit der Eigenschaft

$$p_0\lambda^{(n)}(1-\rho^{(n)}(1-x)) < x$$

für alle $x \in (0, p_0)$ und $p_0$ beliebig nahe an die obere Grenze.

- Tornado Folge
- Rechtsreguläre Folgen (S-1999)
- Funktionaltheoretische Ansätze (Oswald-S, 2000)
- Flachheitsbedingung (S-2000)