

# Spinal符号の実装と評価

石橋 功至 (Koji ISHIBASHI)<sup>1</sup>

<sup>1</sup>: Advanced Wireless Communication Research Center (AWCC),  
The University of Electro-Communications, Tokyo Japan

Email: [koji@awcc.uec.ac.jp](mailto:koji@awcc.uec.ac.jp)

# 本発表の流れ

1. Spinal符号とは？

2. Spinal符号の基礎

- Spinal符号化法・Bubble復号法
- シミュレーションによる特性評価

3. Spinal符号のUSRPを用いた実装・実験評価

4. Spinal符号の特性解析

- Spinal符号の問題点

5. 無線通信とレートレス符号の今後

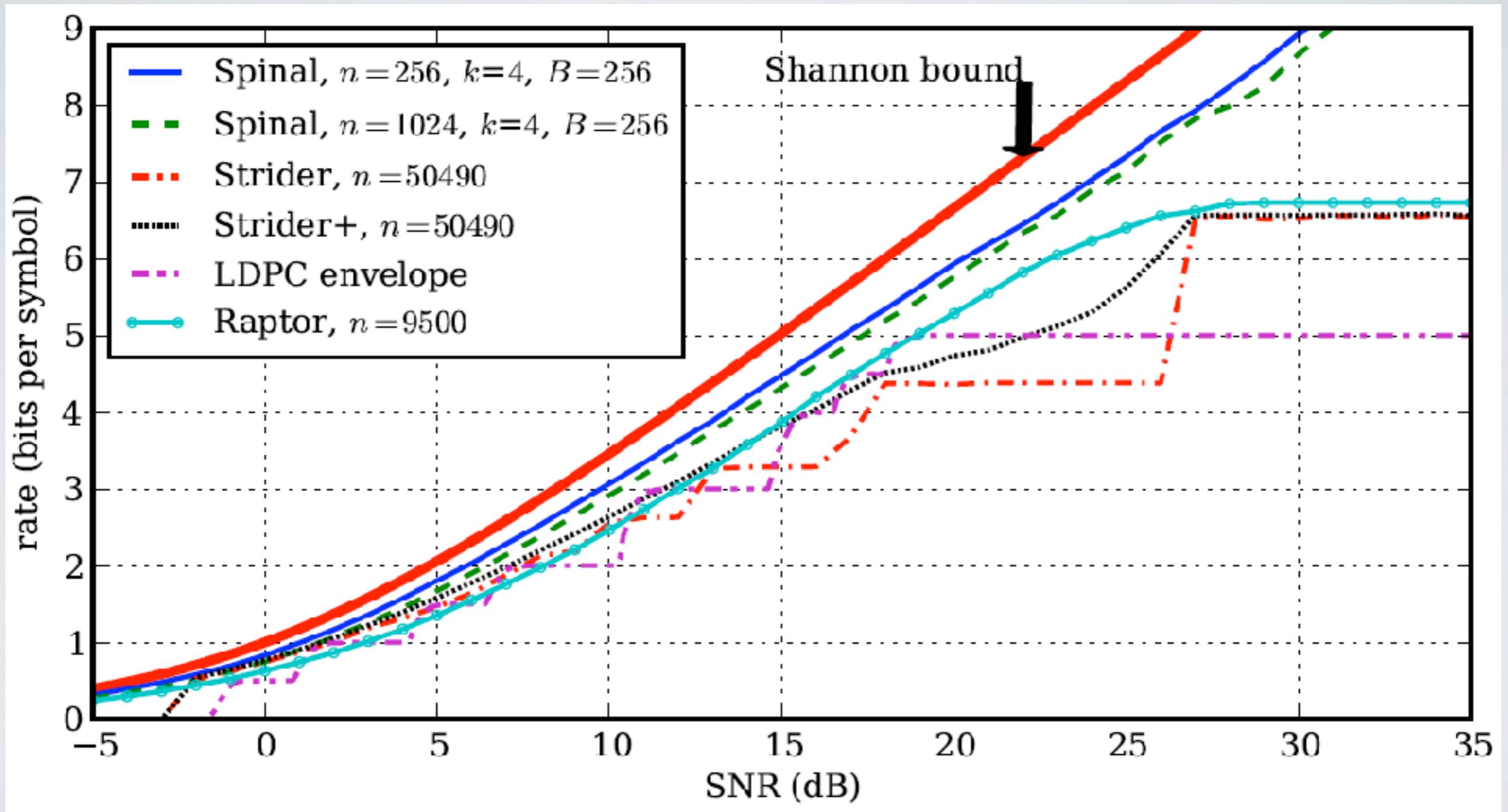
- Spinal符号化を用いた重畳伝送の実装・実験評価

6. まとめ

# Spinal符号とは

- 2012年にMITのグループによって提案されたレートレス符号
  - J. Perry, P. Iannucci, K. Fleming, H. Balakrishanan, and D. Shah, “Spinal codes,” Proc. ACM SIGCOMM, Helsinki, Finland, August, 2012.
  - 引用回数 31件 (2014年9月現在 - Google Scholar調べ)
- Spinal符号の長所
  - ハッシュ関数を用いた簡易な符号化法
  - Mアルゴリズムに基づく低演算復号法
  - 通信路容量に漸近する高い平均伝送レート特性

# Spinal符号とは



# 気になった点

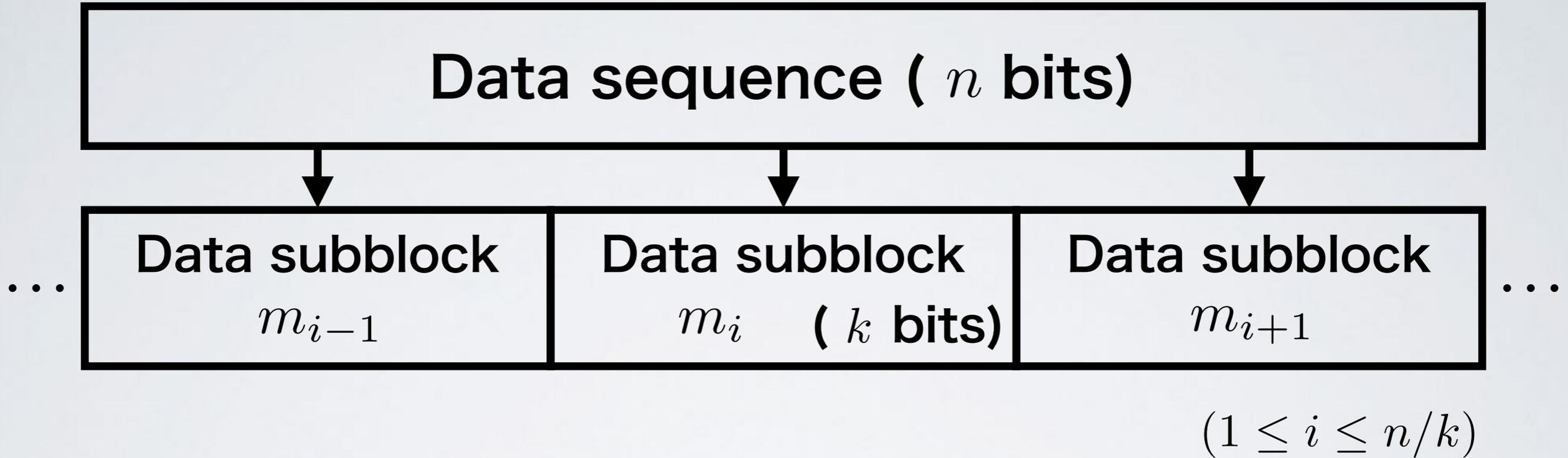
- 著者らの主張：

- ハッシュ関数を用いることにより復号可能なランダム符号化を実現したことによって通信路容量に漸近した [SIGCOMM12]
- 符号が無限であれば通信路容量を達成可能 [ArXive12]

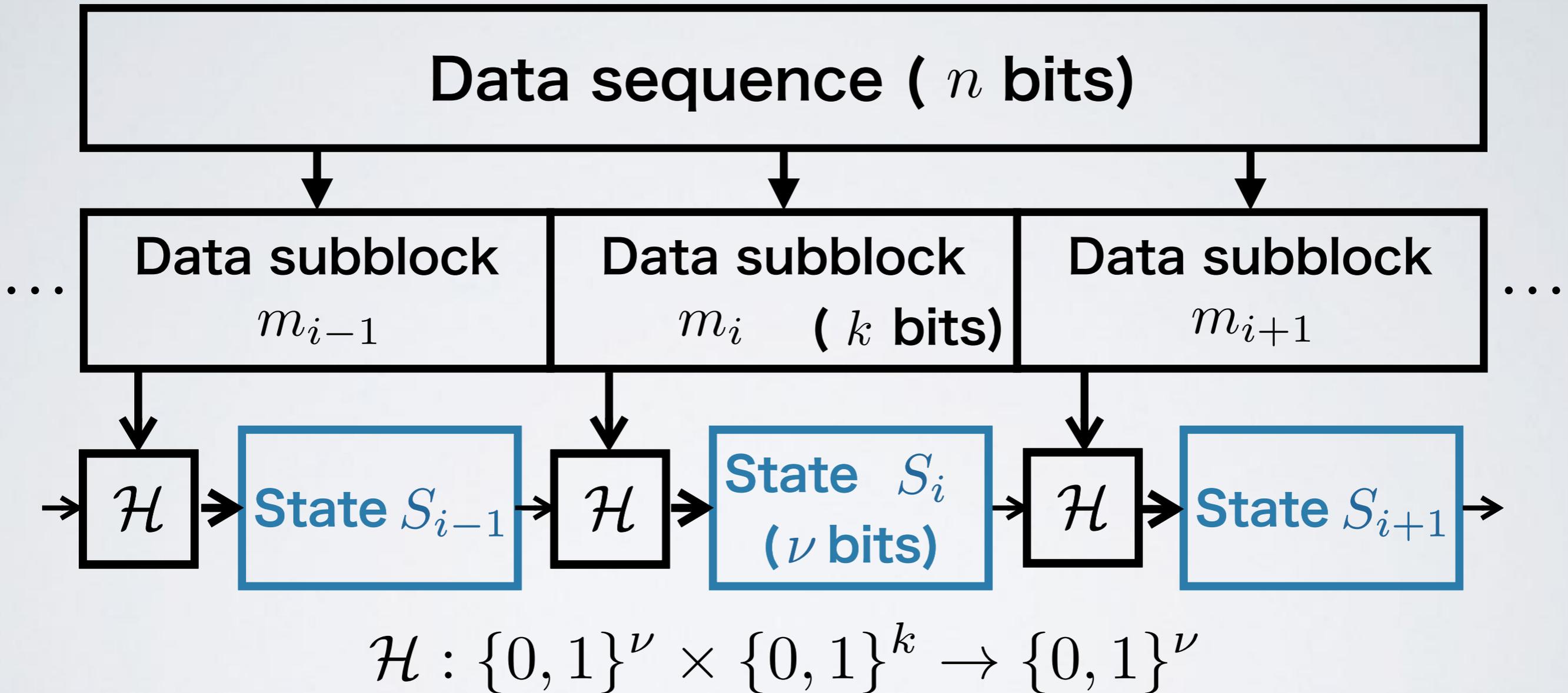
- 気になる点：

- ランダムな符号であれば性能が良い
  - 有限符号長のものに対しても成立するのか？
- 評価は平均データレートのみ
  - 任意の誤り率は達成可能？
- 変調が矩形QAM変調
  - 矩形QAM変調の平均相互情報量とのギャップは？

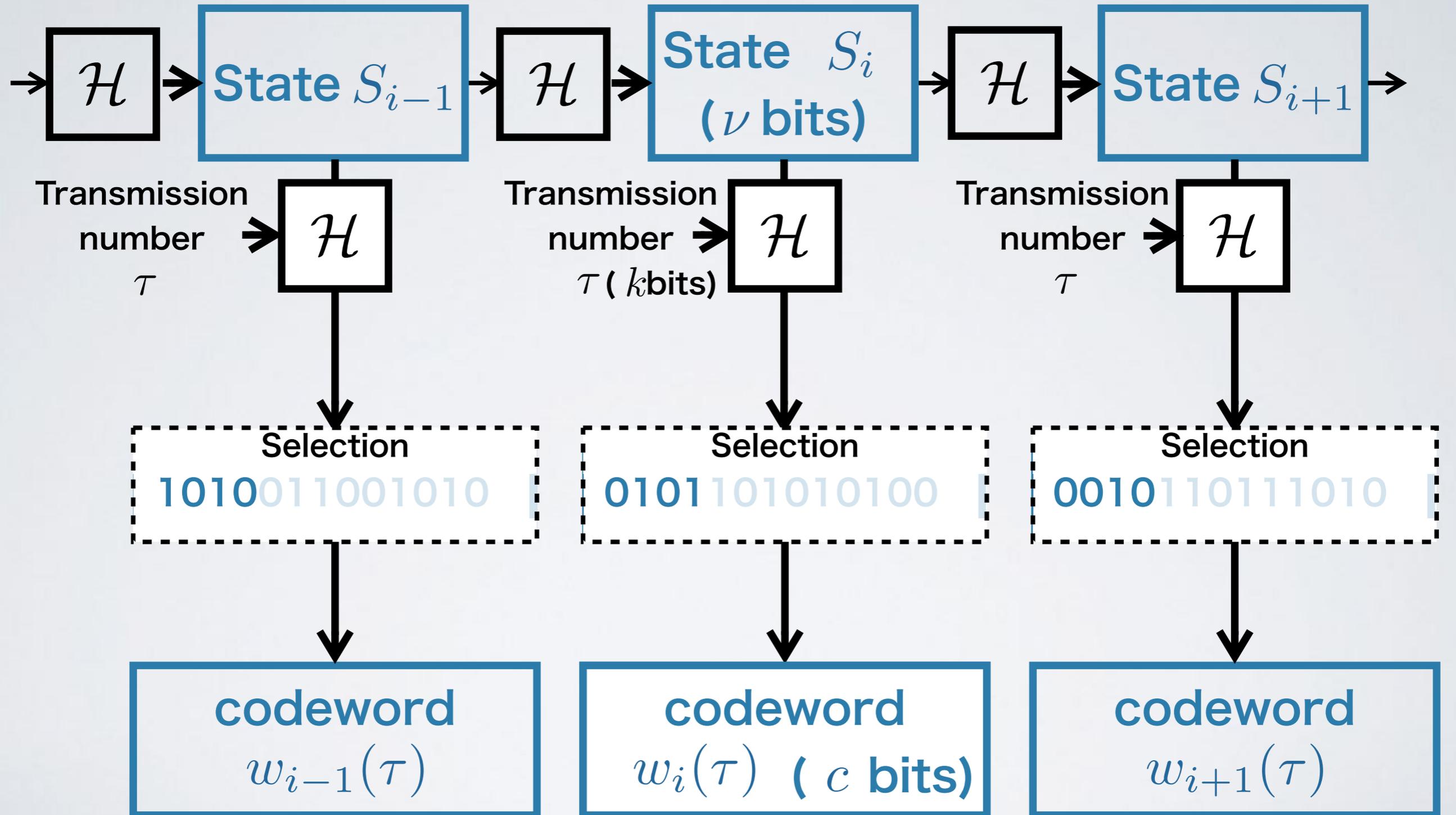
# Spinal符号化



# Spinal符号化

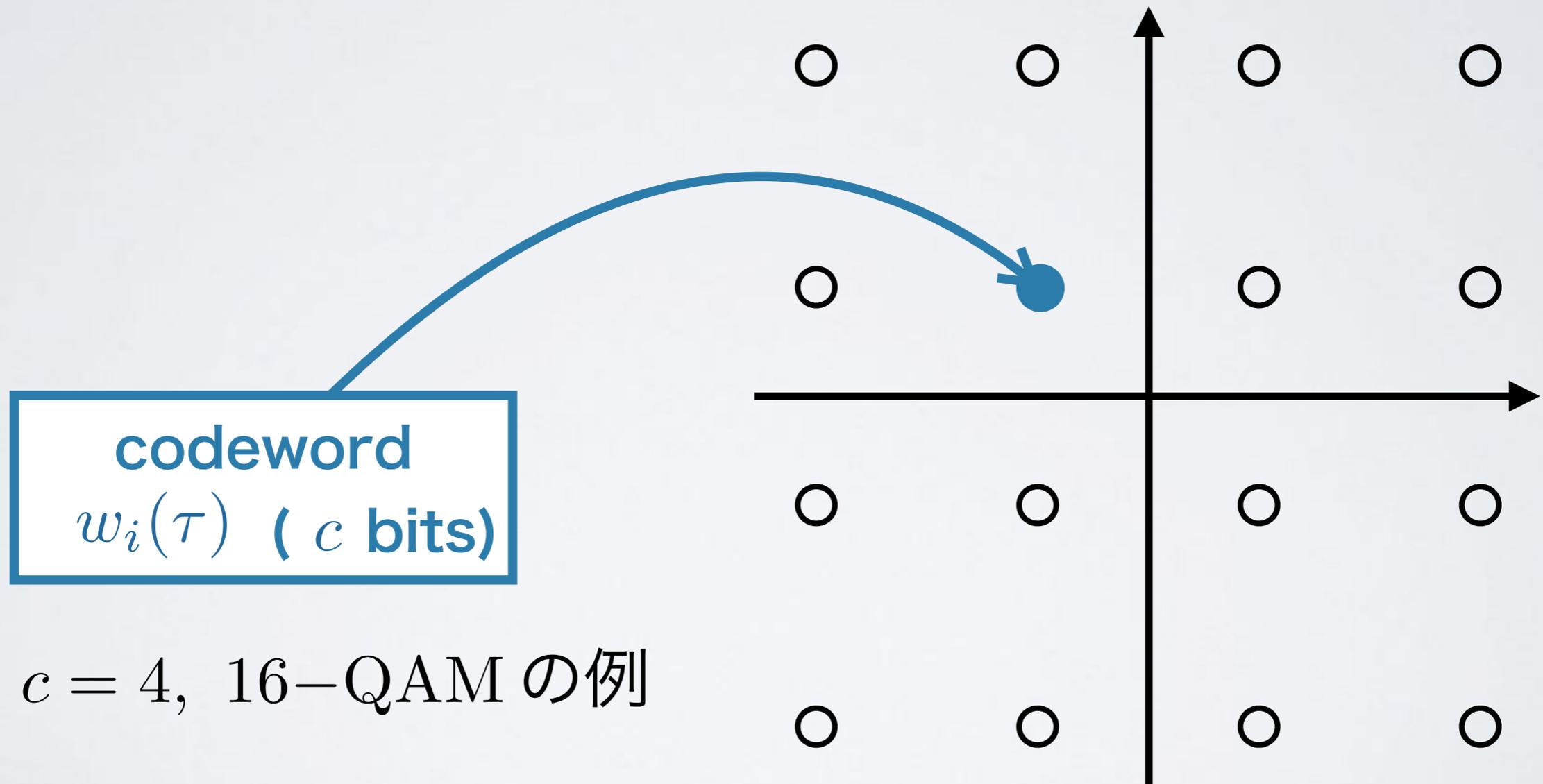


# Spinal符号化



# 変調と送信

- 符号語を  $|\mathcal{X}| = 2^c$  の変調点空間  $\mathcal{X}$  にマッピングし、送信する

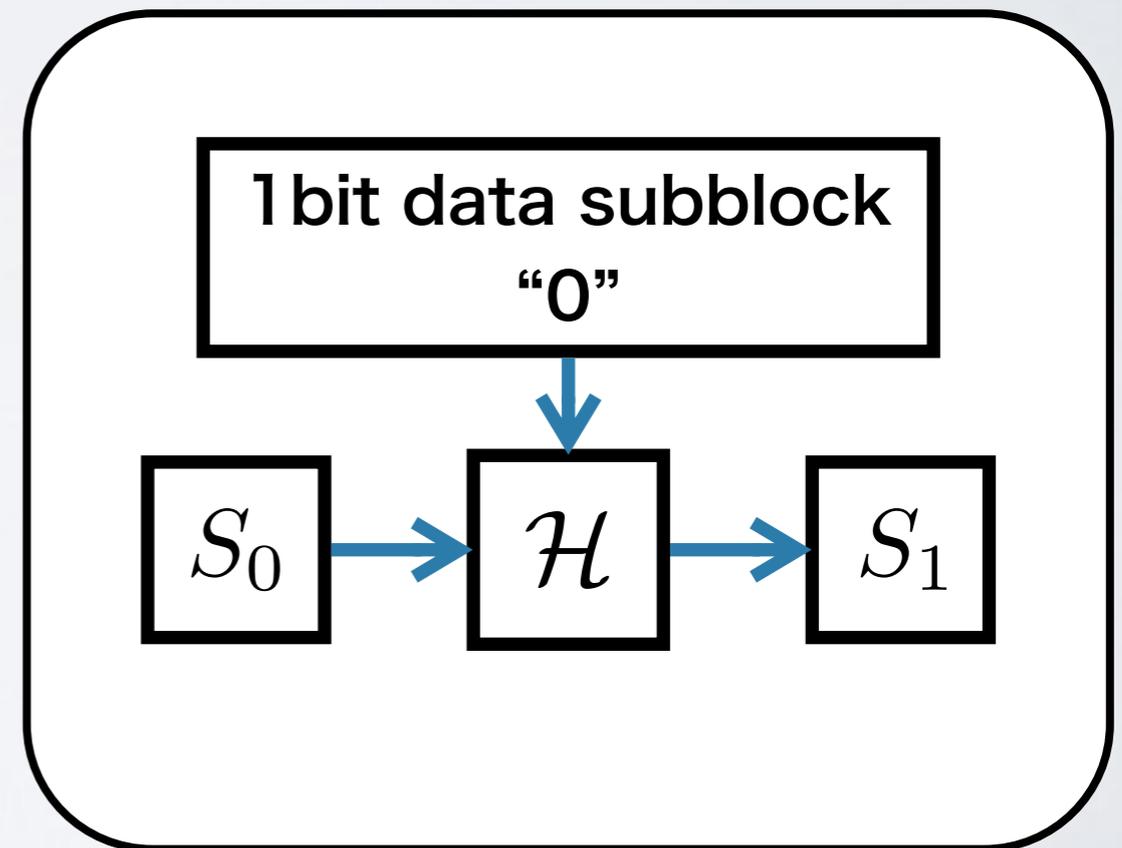
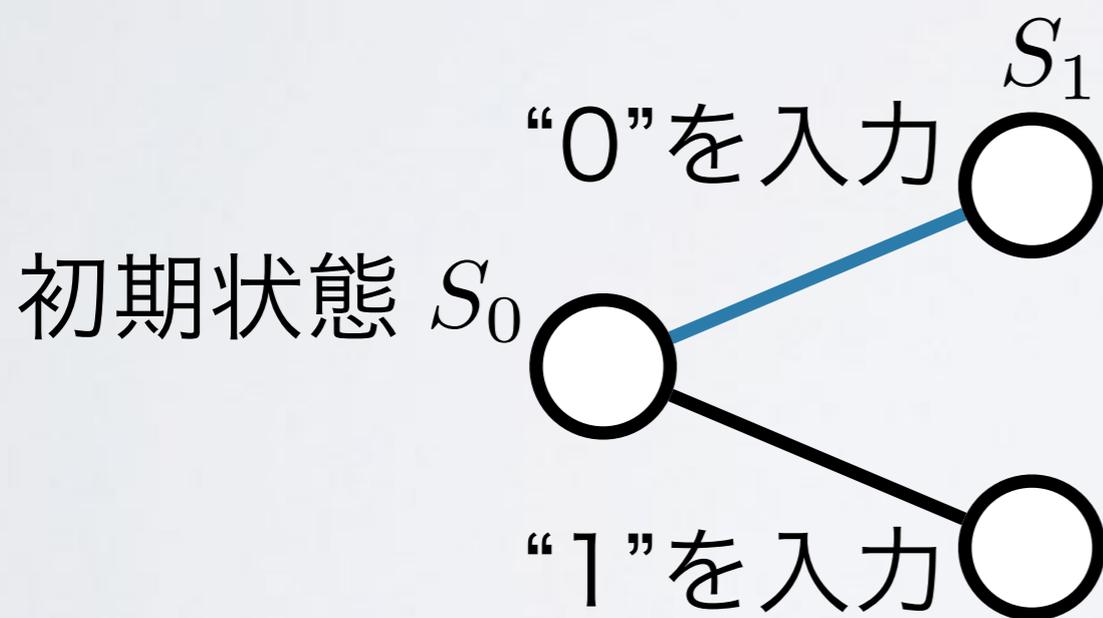


# バブル復号

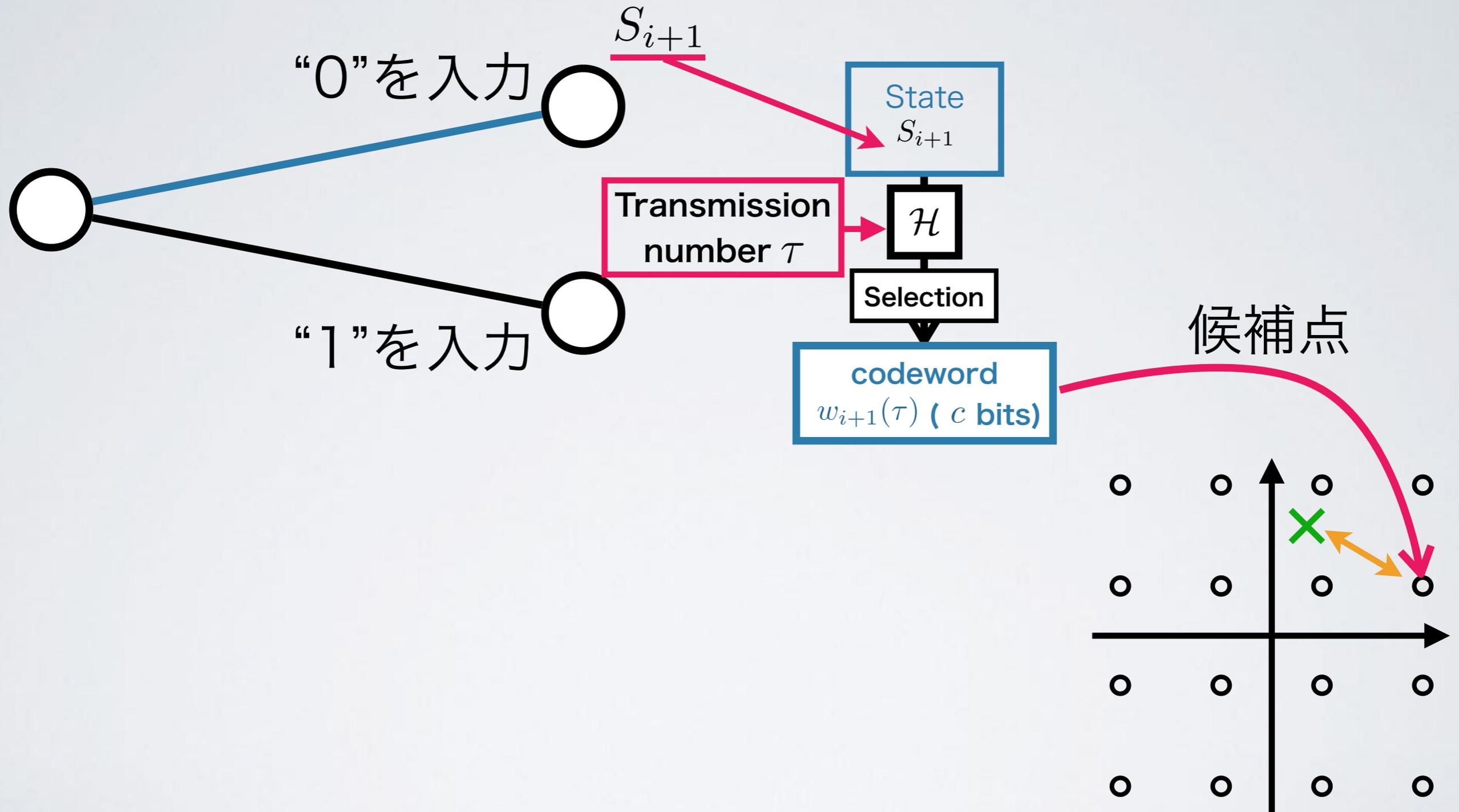
- 文献[SIGCOMM12]で提案されている復号法
- 最尤復号の近似アルゴリズム
  - ビタビアルゴリズムによる最尤復号が最適
  - 32ビットハッシュの場合、状態数が約4億3000万
- ビーム幅と深さと呼ばれる変数によって候補状態の絞り込みを行うことで演算量を低減（M-Algorithmの一種）

# バブル復号

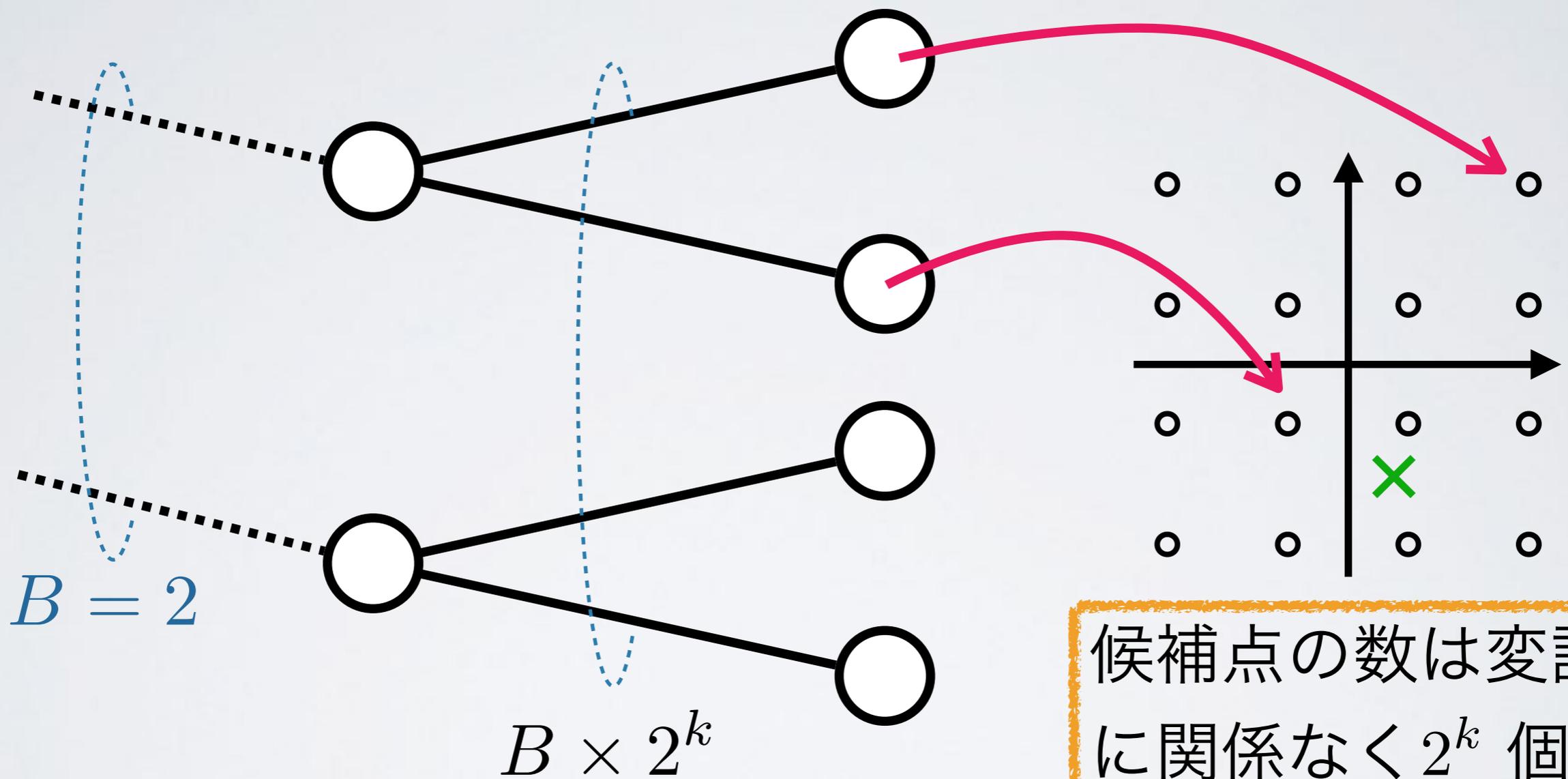
- 初期状態は送受信機間で共有されているとする
- ハッシュ関数に入力される情報サブブロックを枝、状態を節とする木で状態の遷移を表現し、復号



# 枝メトリック計算

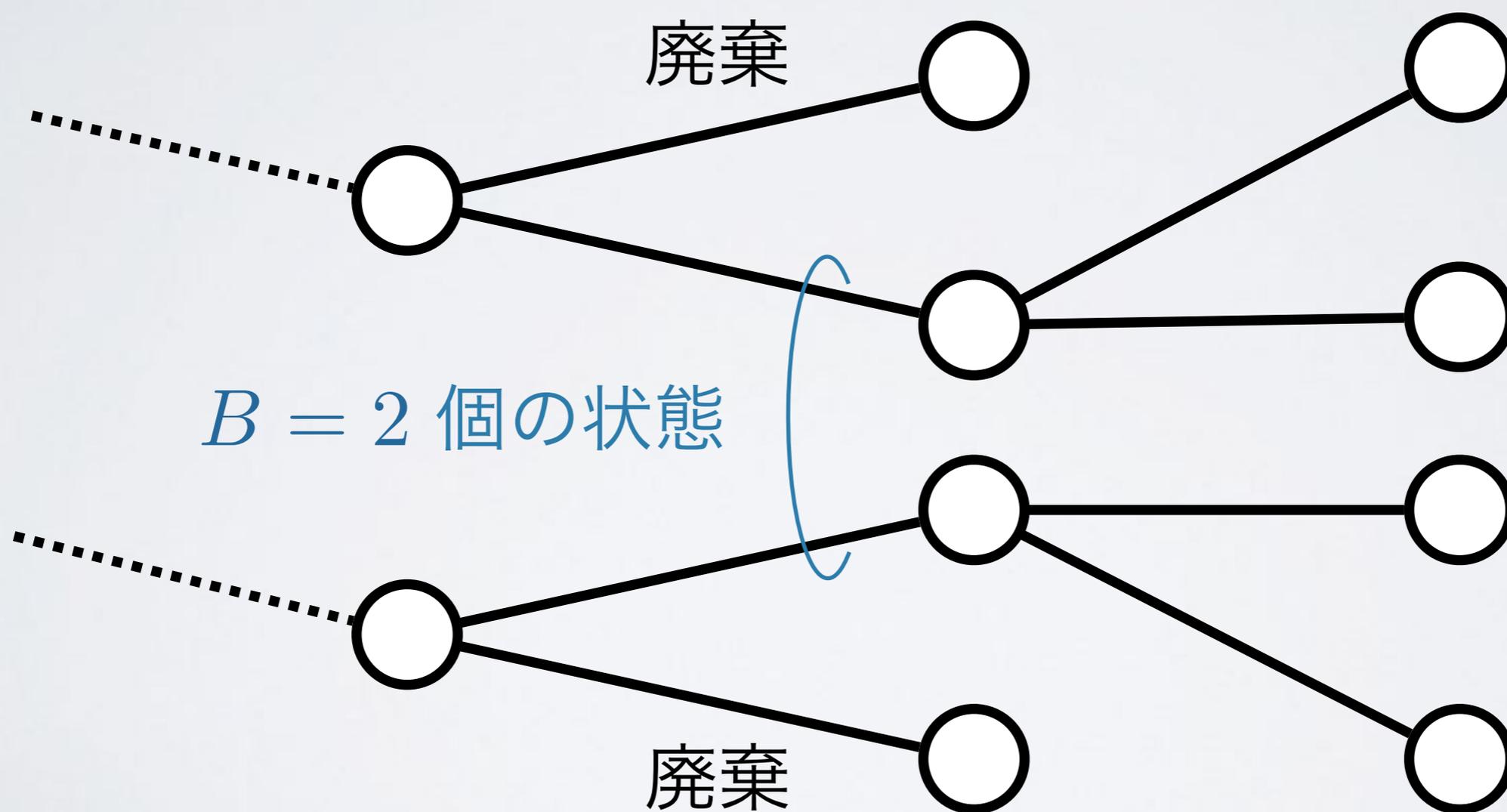


# 枝メトリックの計算



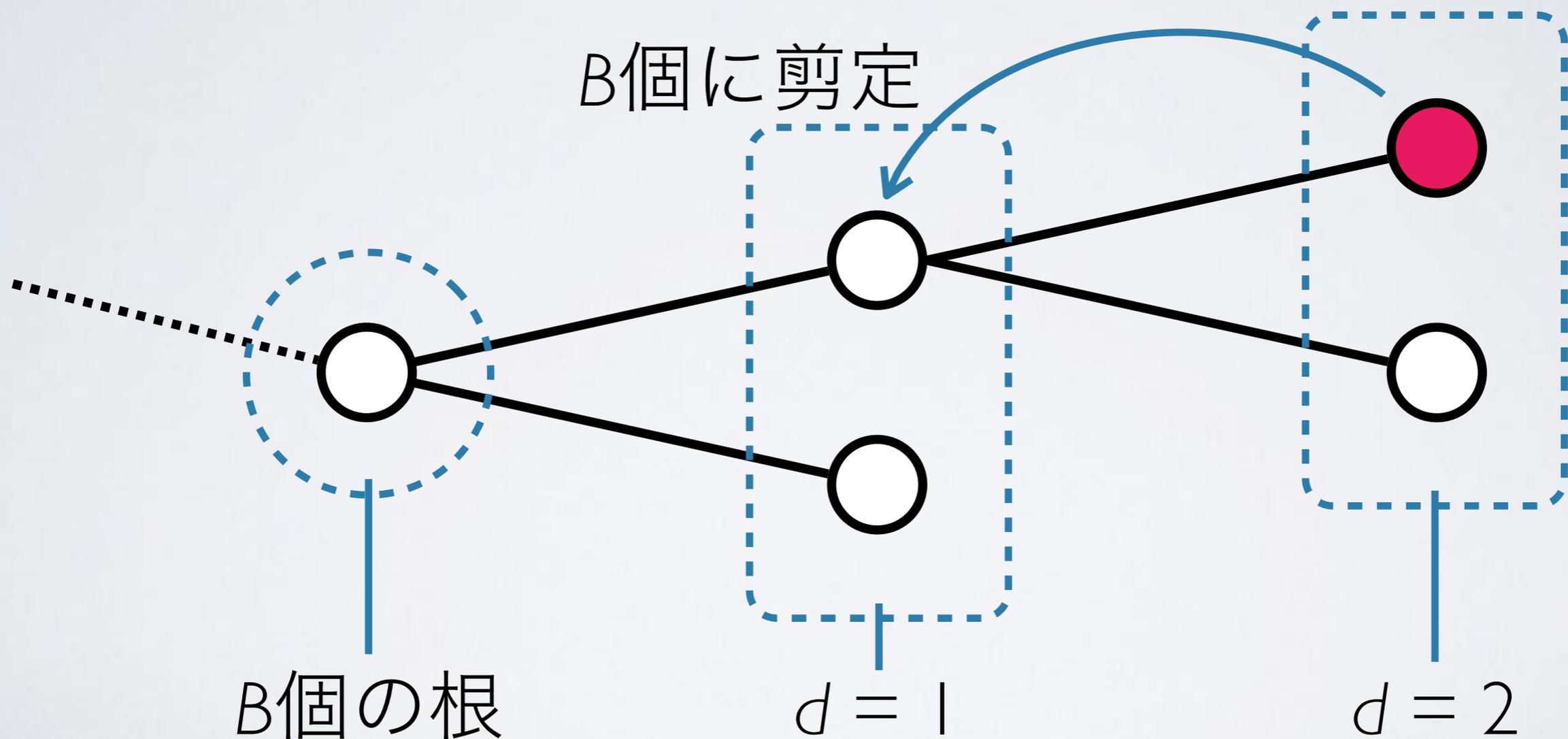
# パスの剪定：ビーム幅

- 拡張された  $B \times 2^k$  本のパスから，パスメトリックの小さい  $B$  本のみを残し，他のパスをすべて廃棄する



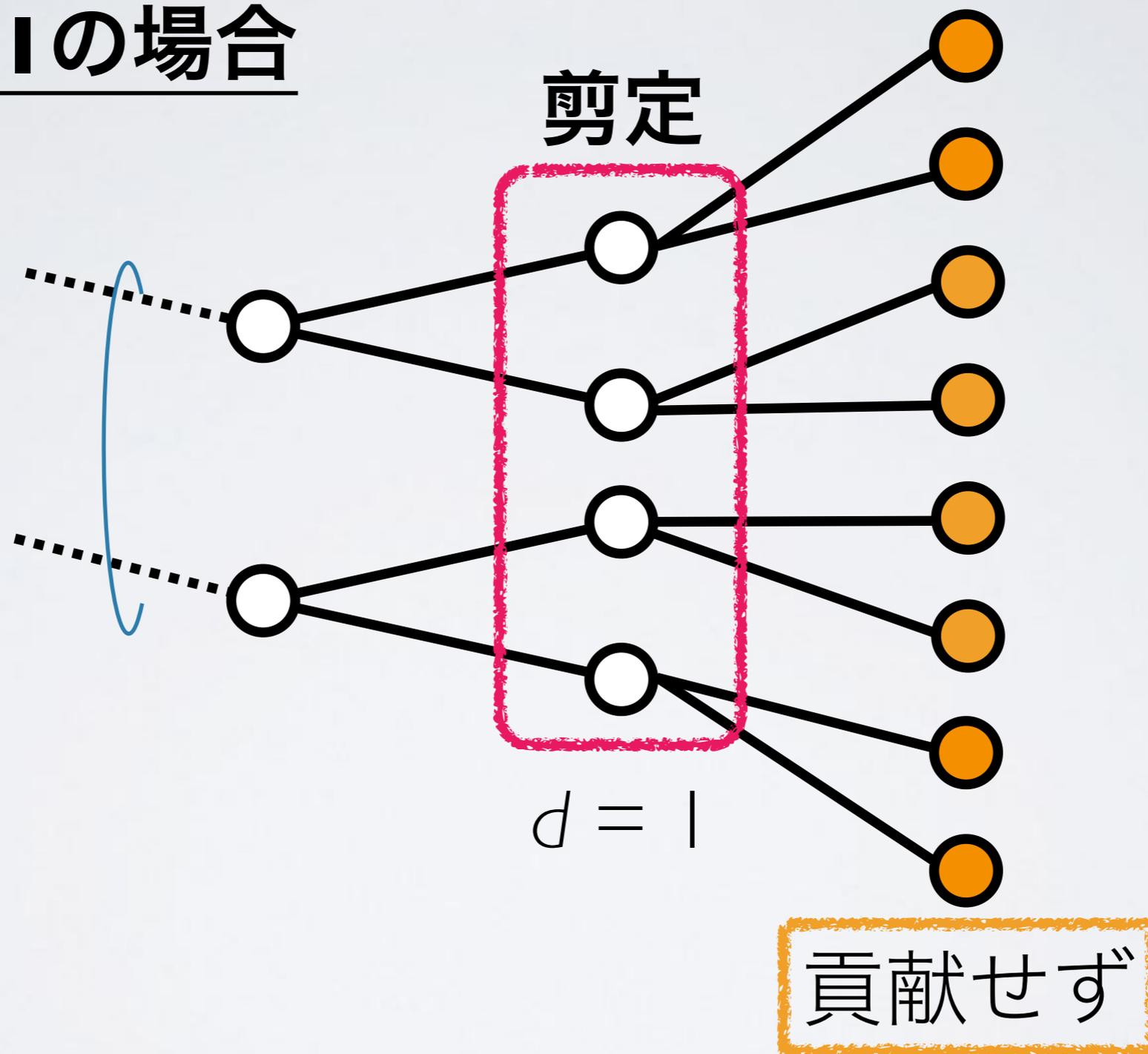
# パスの剪定：深さ

- 拡張された  $B \times 2^k$  個の状態からさらに  $(d-1)$  時点、枝を伸ばし、その時点でもっとも高い尤度を持ったものを、剪定に用いる



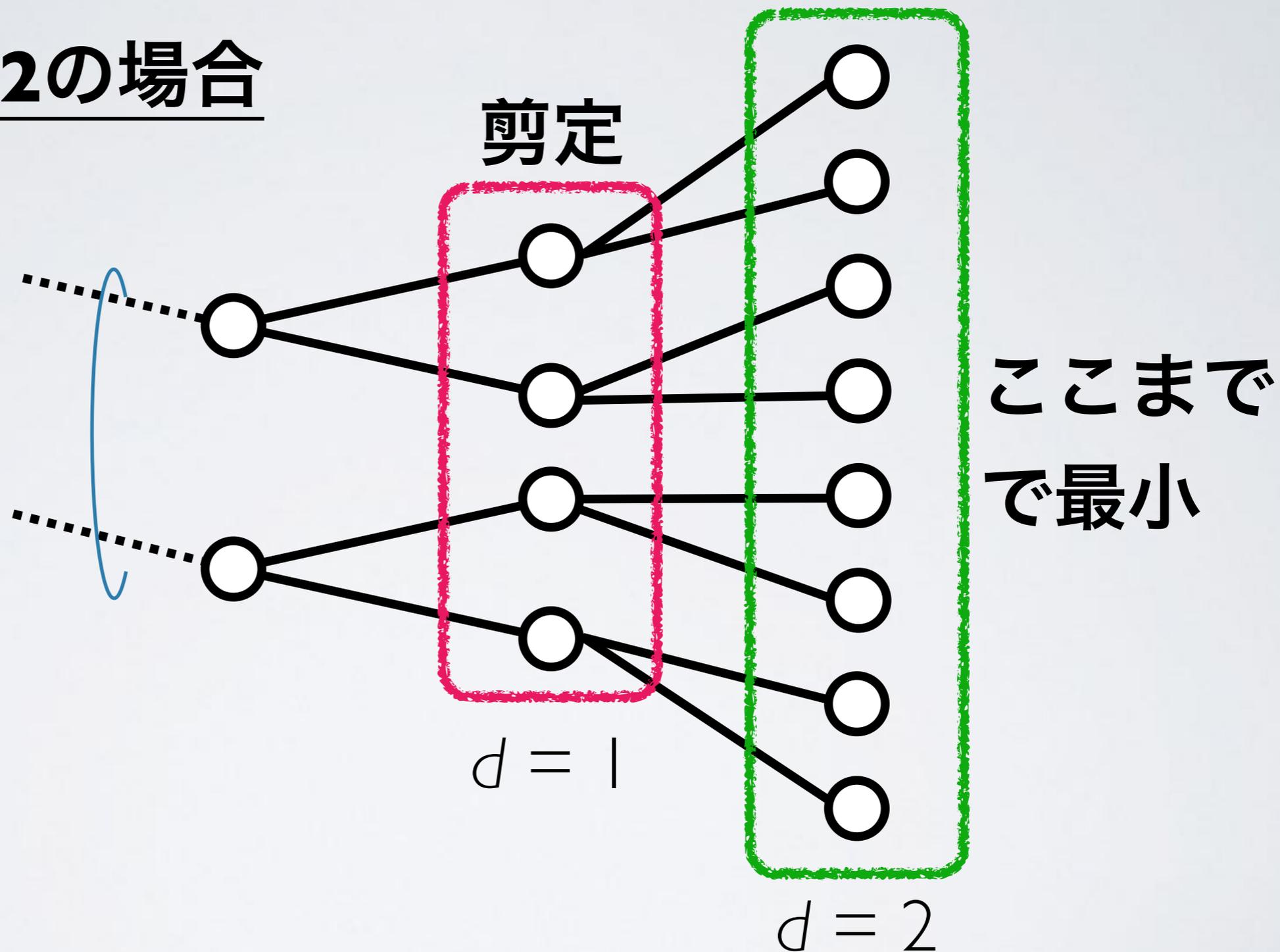
# 深さとビーム幅

$B = 2, d = 1$ の場合



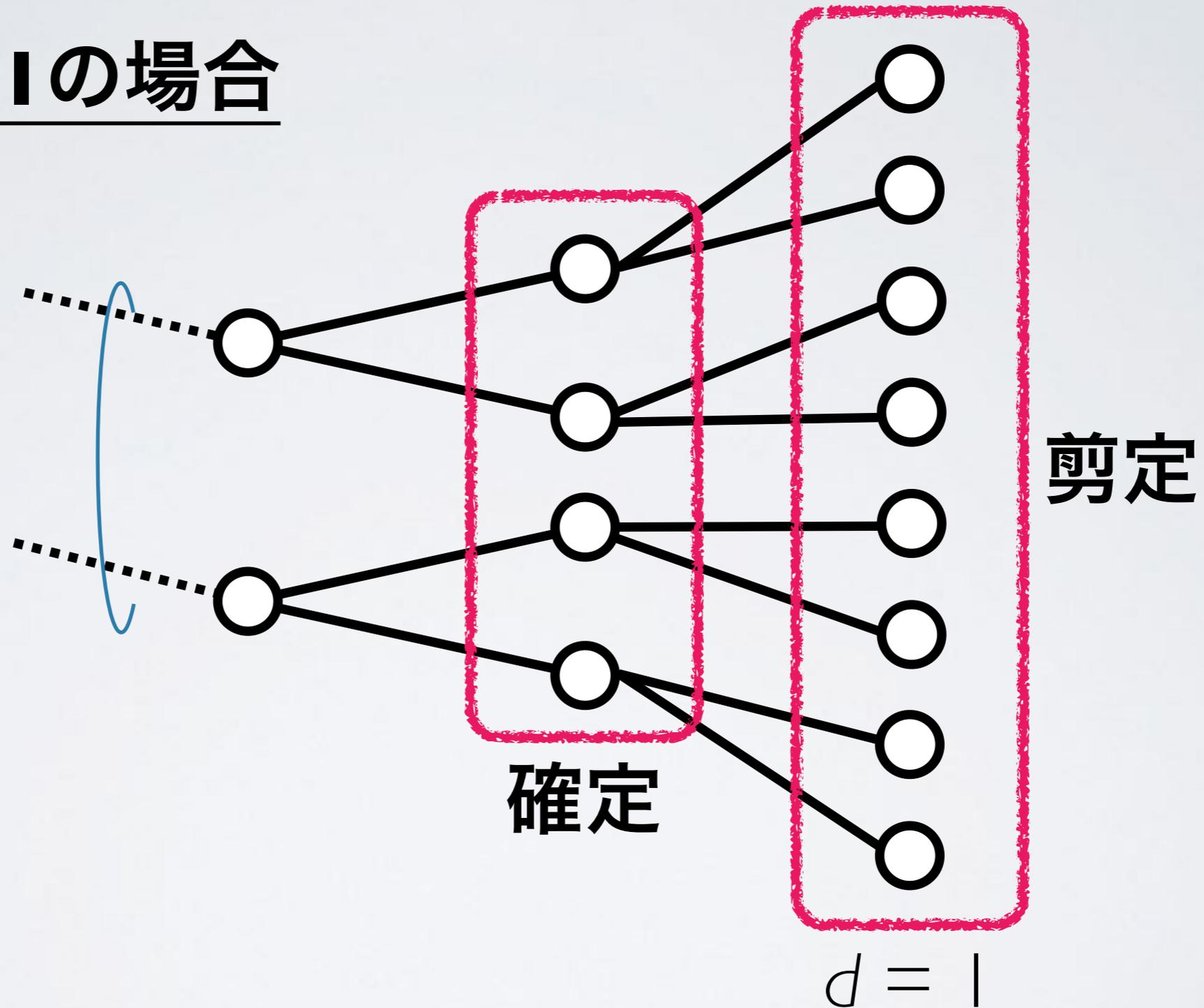
# 深さとビーム幅

$B = 2, d = 2$ の場合



# 深さとビーム幅

$B = 4, d = 1$ の場合



# 深さとビーム幅

- 深さを増やすこととビーム幅を増やすことは本質的に同じ（結局どこまでを考慮して剪定するか）
- 計算量とメモリ量：
  - 剪定時に  $2^k B d$  回だけメトリック計算が必要
  - メモリ量は  $2^k (B + d - 1)$
  - 変調点や状態数に対して不感（実装上は重要）

# 計算機シミュレーション

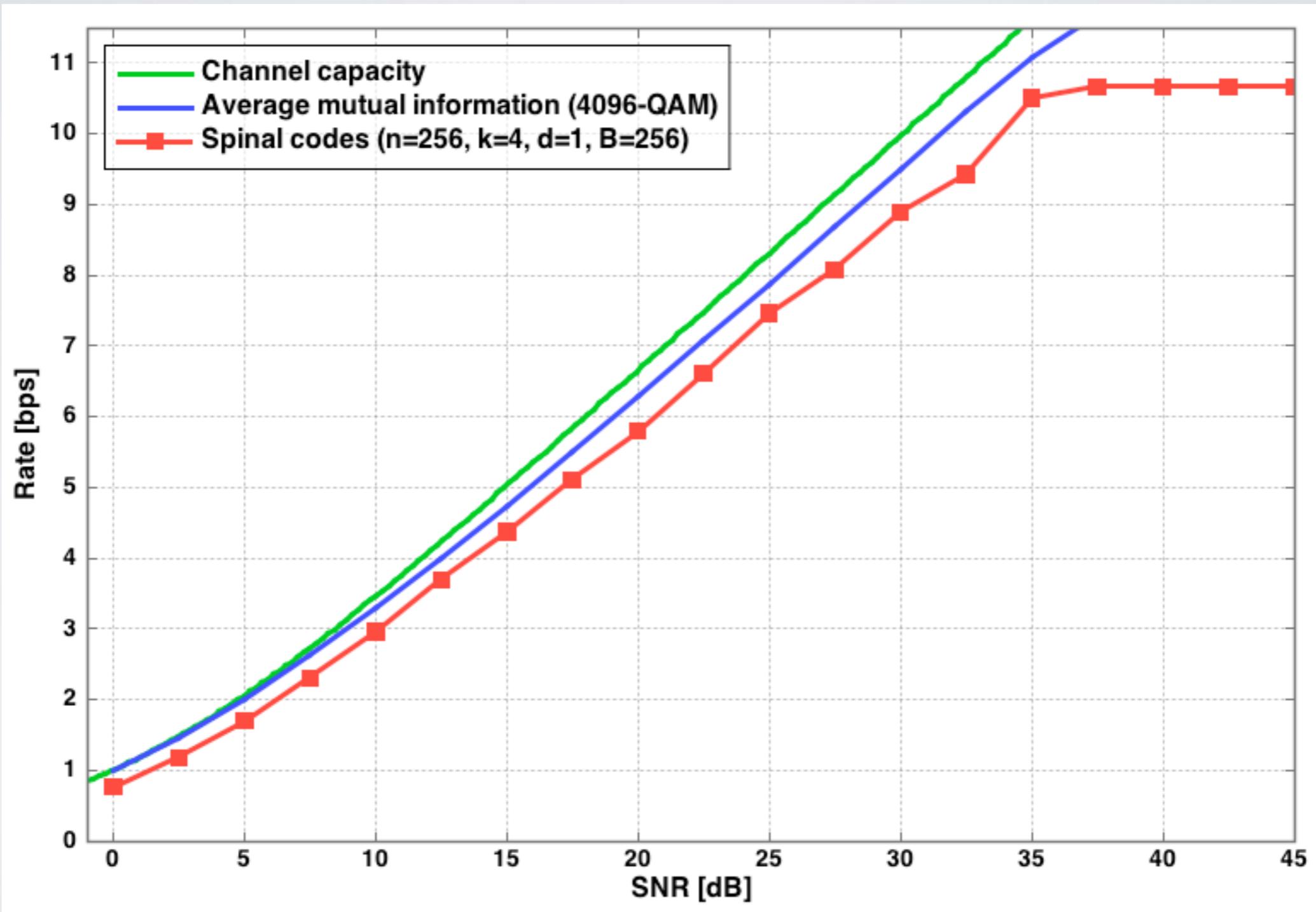
## Simulation parameters

---

❖ Data length $n$	256 [bits]
❖ Data subblock length $k$	4 [bits]
❖ Codeword length $c$	12 [bits]
❖ Modulation order	4096-QAM
❖ Hash value size $\nu$	32 [bits]
❖ Beam-width	256
❖ Depth	1

---

# 伝送レート特性



# USRPを用いた実装

- Spinal符号を実装して実環境評価 (静大 猿渡先生との共同)
- ソフトウェア無線機：USRP N200
  - 30万程度で一式購入可能
  - National Instruments(NI)から免許付きで購入が可能
  - ドーターボードの変更により対応周波数が可変
  - ハードウェアドライバ：Universal Hardware Driver (UHD)
    - 誰でも開発が可能 (C++とPython)
- コンピュータサイエンス系の学会ではUSRPを使って実装・評価することが常識に (GitHub上にソースコード有)



# USRP実機



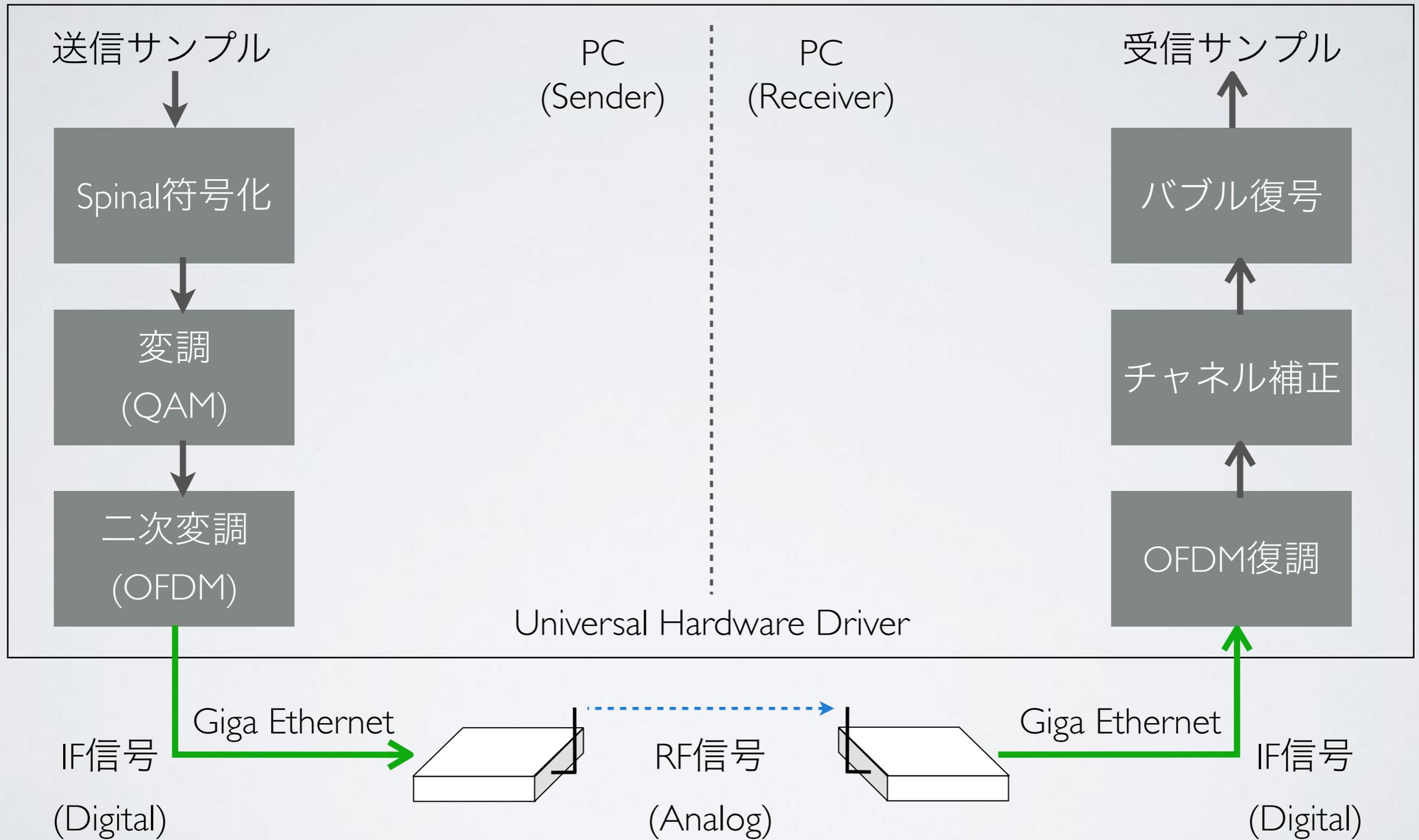
# USRP N200仕様

USRP Model	Interface	Total Host BW (MSPS 16b/8b)	Daughterboard Slots	ADC Resolution (bits)	ADC Rate (MSPS)	DAC Resolution (bits)	DAC Rate (MSPS)	MIMO Capable	Internal GPS Disciplined Oscillator (Optional)	1 PPS/Ref Inputs
N210	Gig. Eth.	50/100	1	14	100	16	400	Yes	Yes	Yes
N200	Gig. Eth.	50/100	1	14	100	16	400	Yes	Yes	Yes
B100	USB 2.0	8/16	1	12	64	14	128	No	No	Yes
USRP1	USB 2.0	8/*	2	12	64	14	128	Yes	No	No
E100	Embedded	8/16	1	12	64	14	128	No	Yes	Yes
E110	Embedded	8/16	1	12	64	14	128	No	Yes	Yes

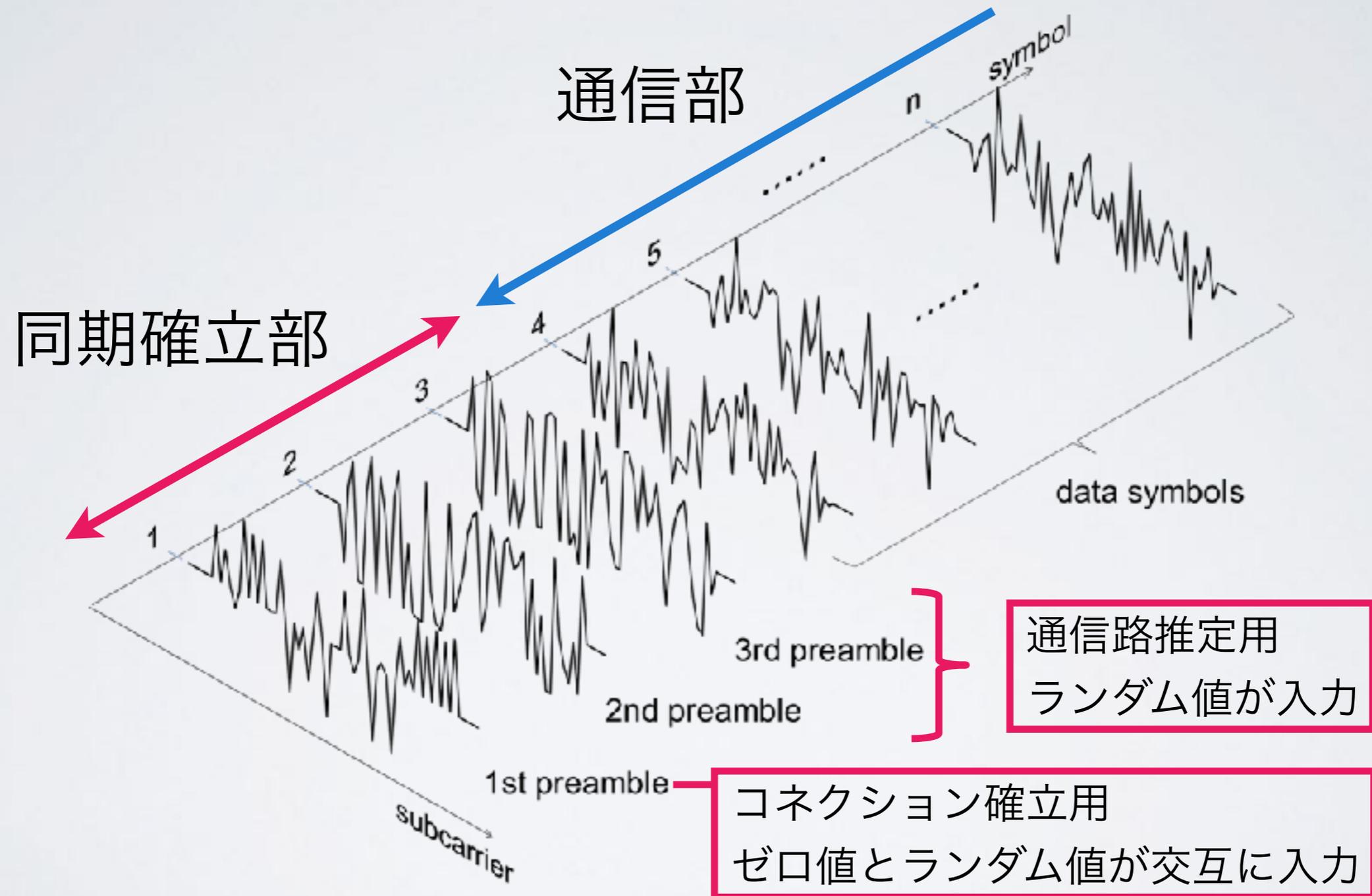
Table 2 - USRP Characteristics by Model

※ Ettus Application Notesより抜粋

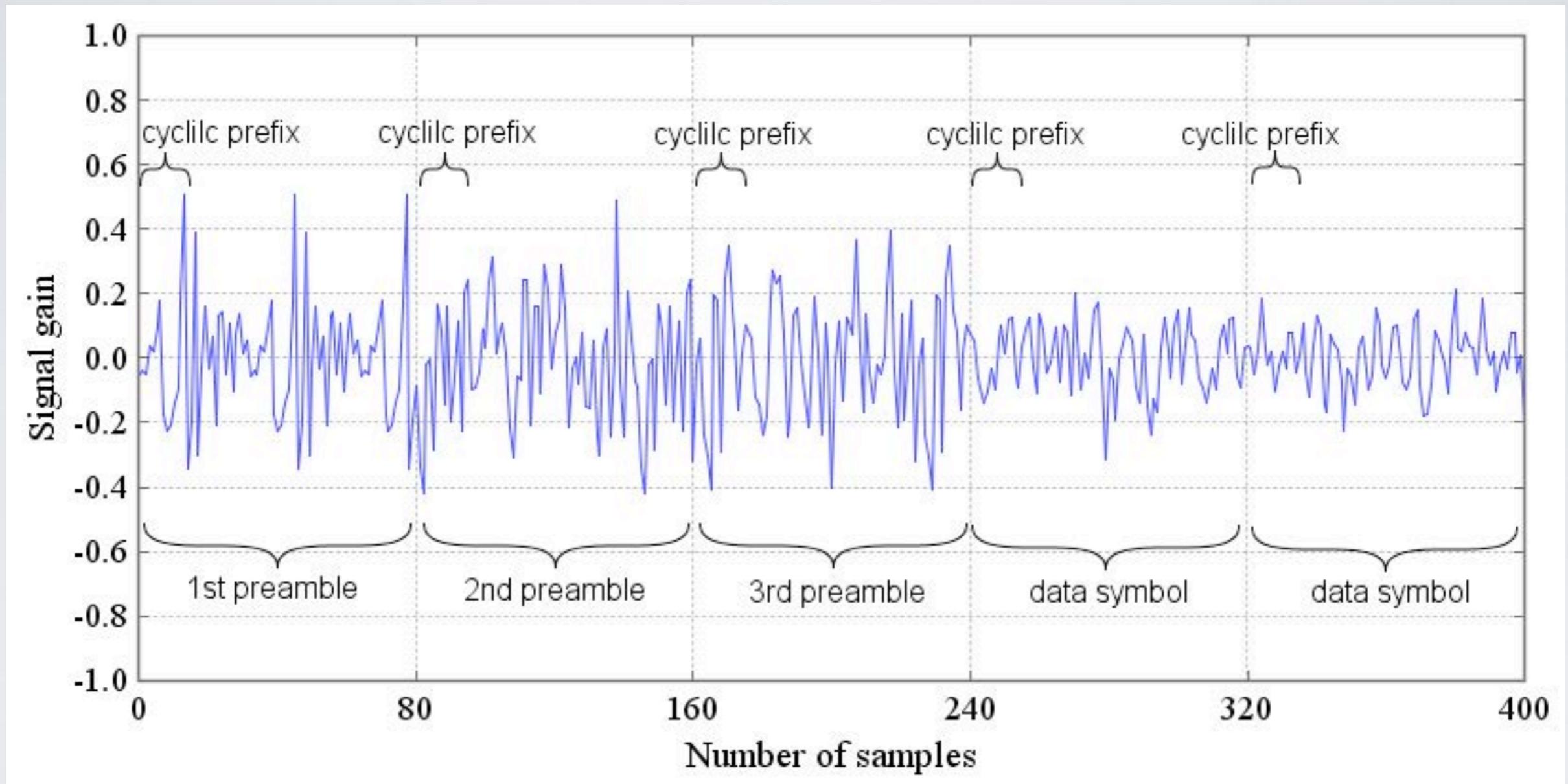
# システム構成



# 同期用フレームフォーマット

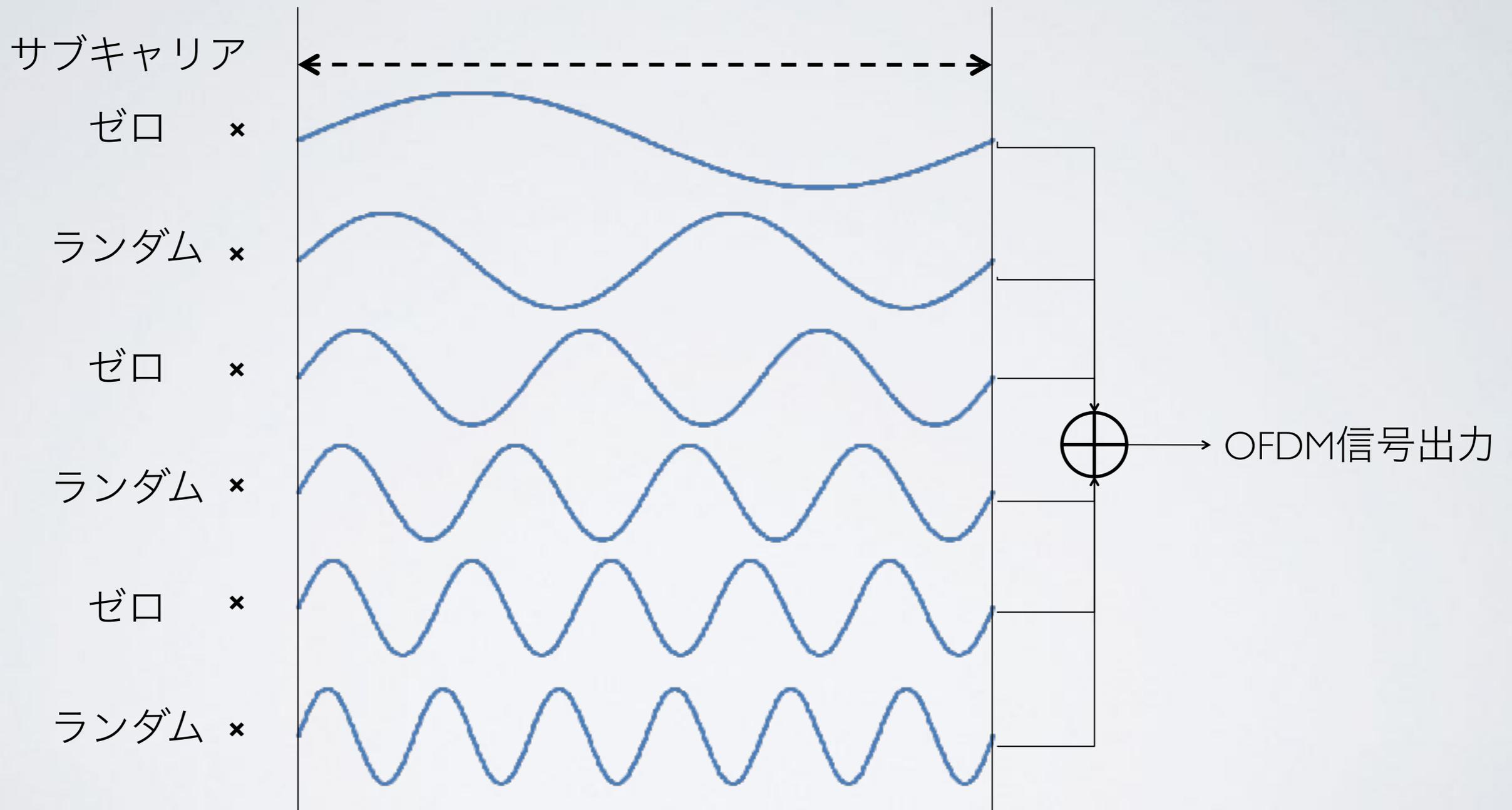


# 送信信号例



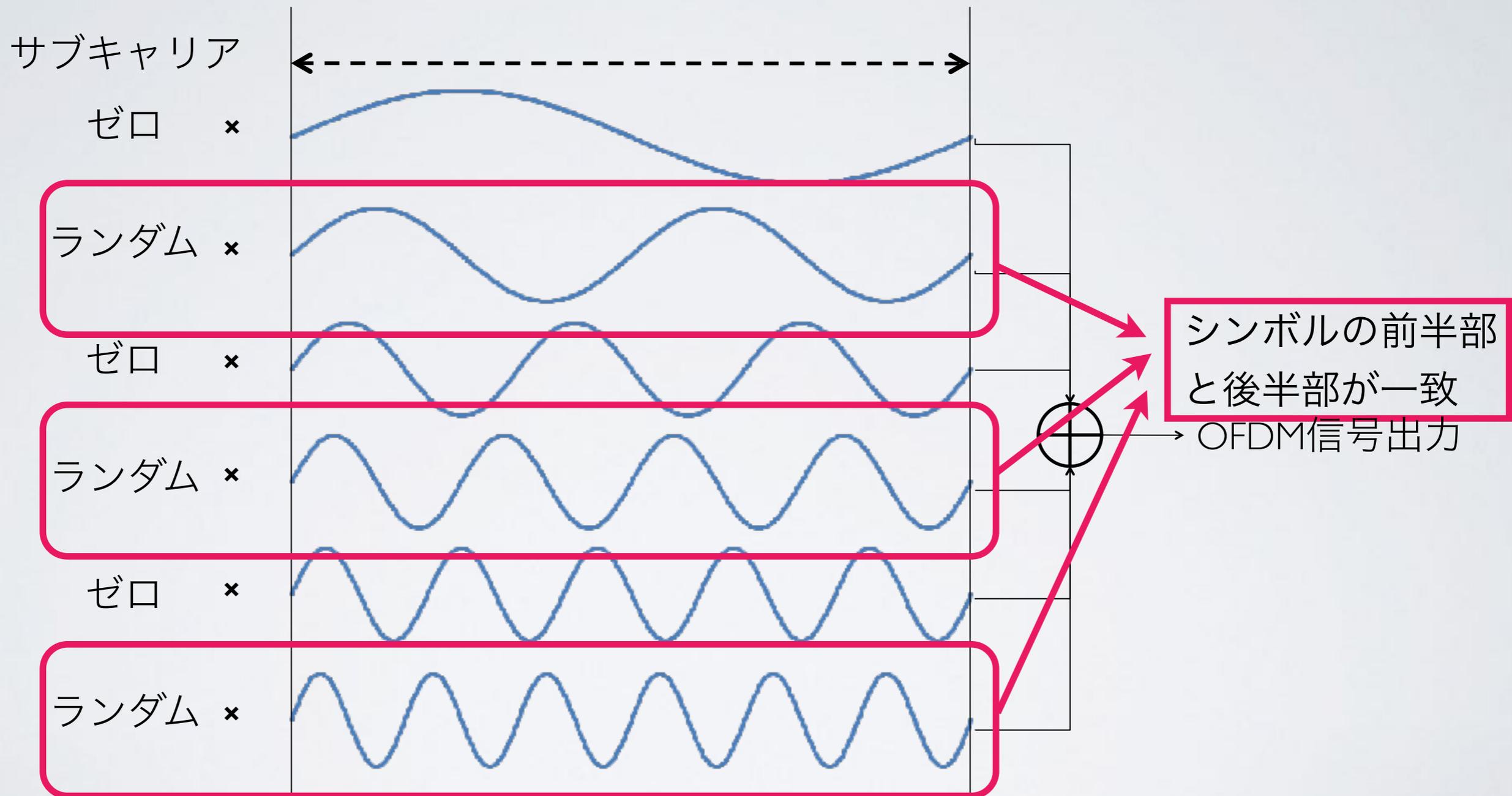
# フレーム同期

- 1つめのプリアンブルを利用



# フレーム同期

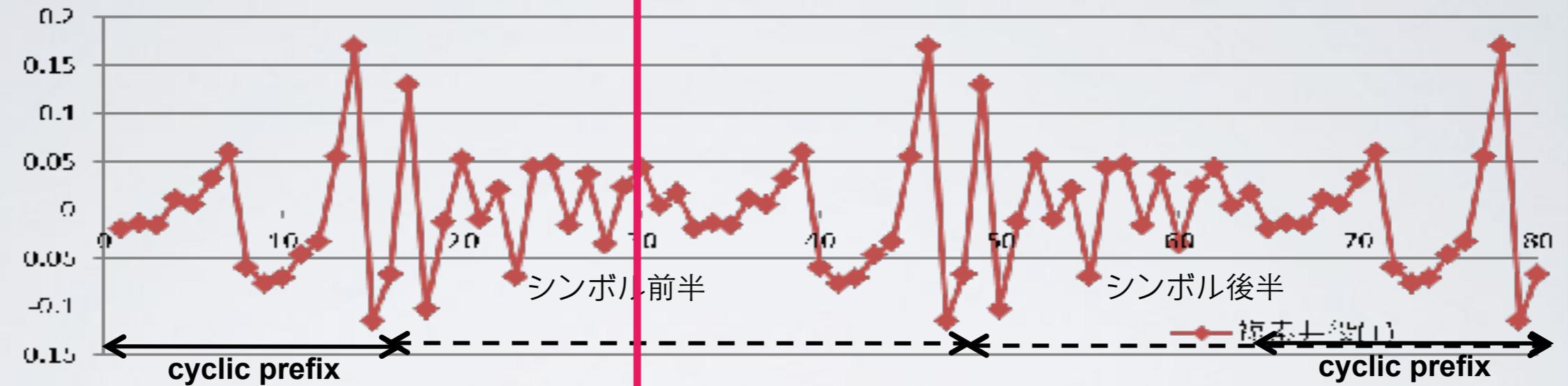
- 1つめのプリアンブルを利用



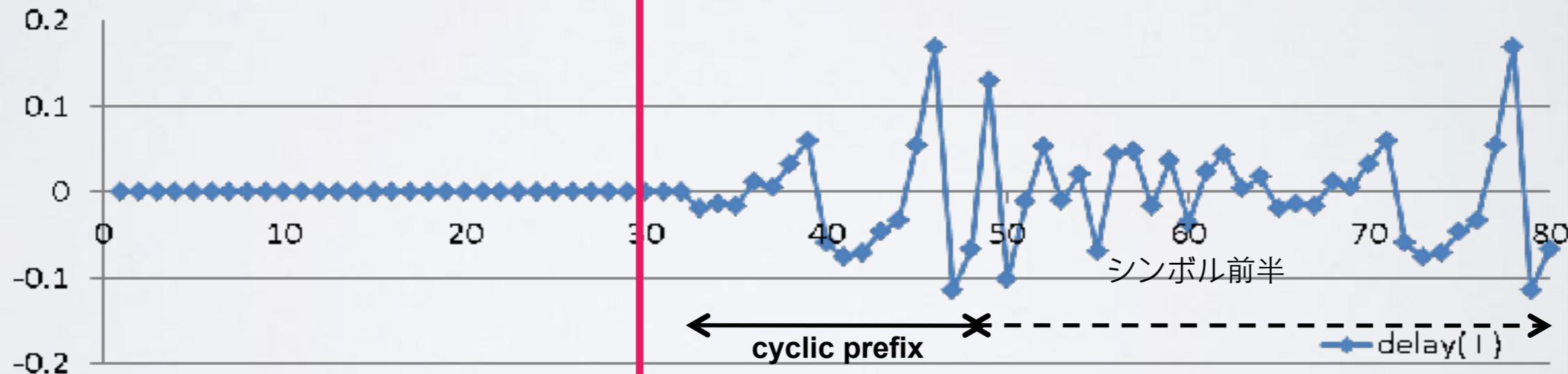
# フレーム同期

1つ目のプリアンブル部分

受信データ



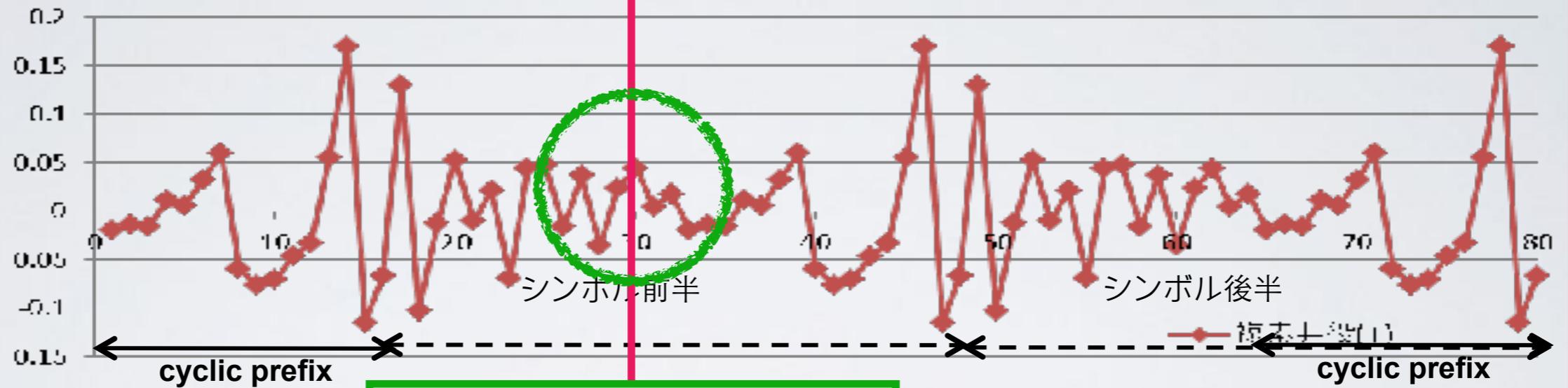
半波長遅延



# CFO補正

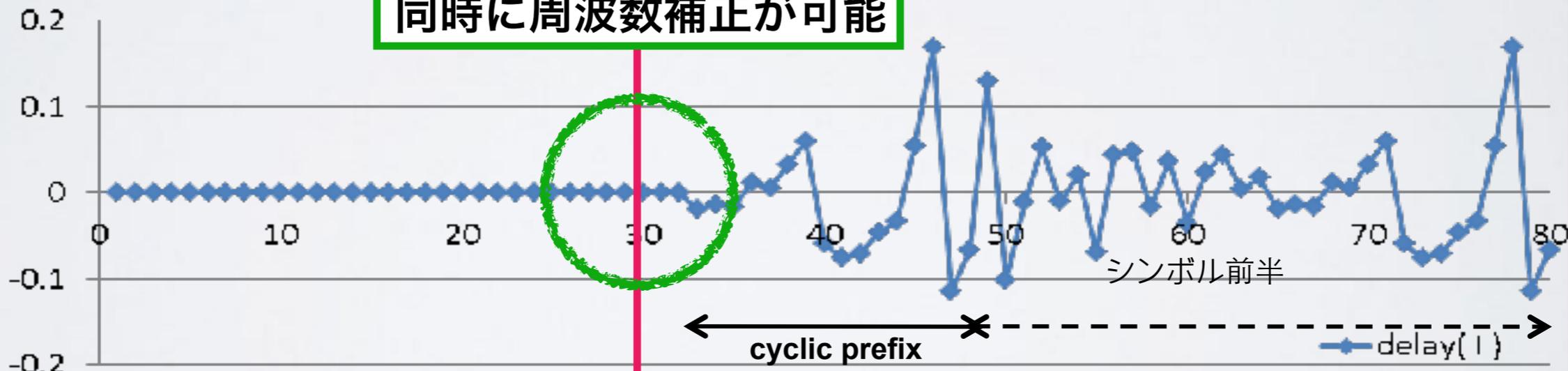
1つ目のプリアンブル部分

受信データ



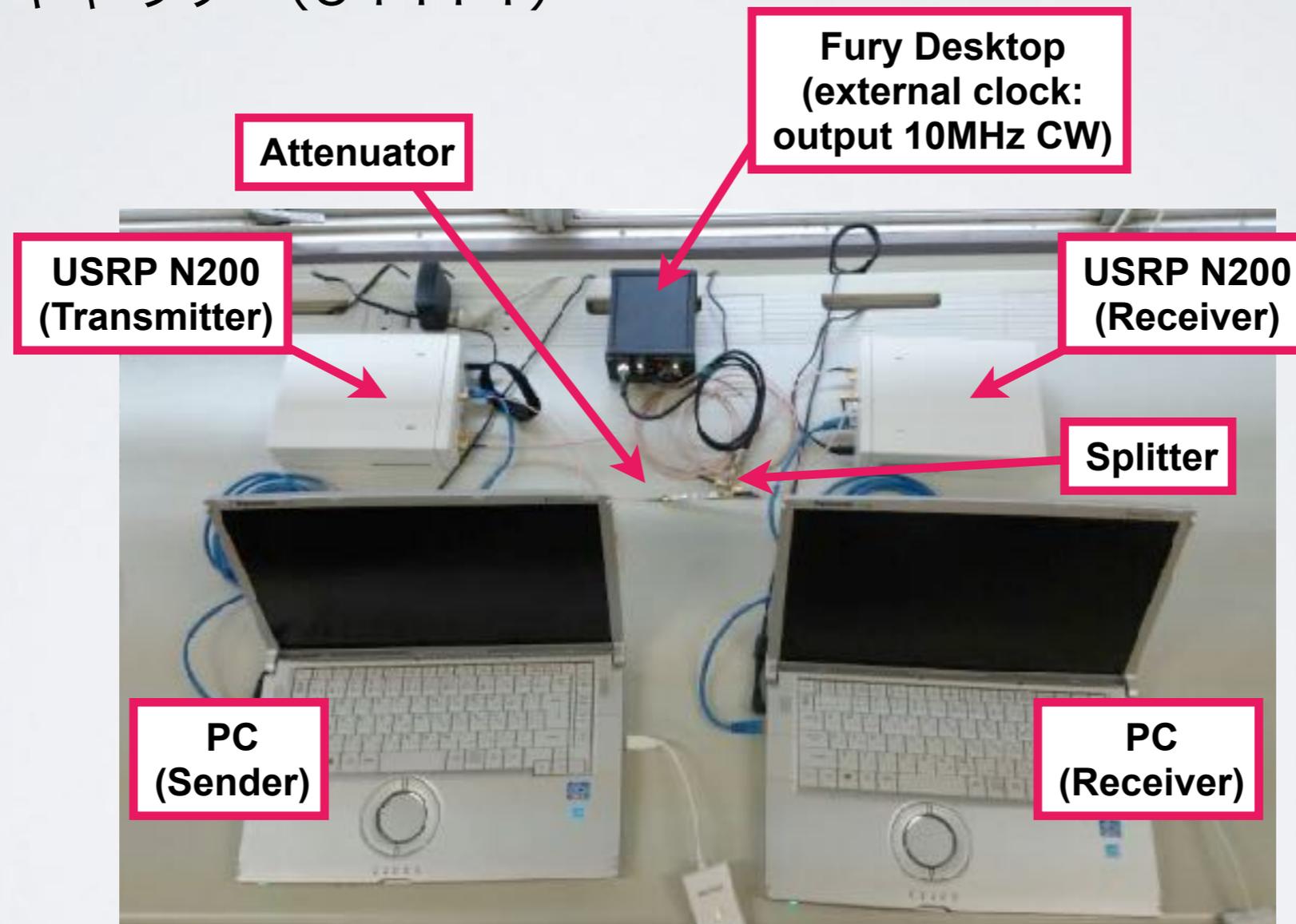
同時に周波数補正が可能

半波長遅延

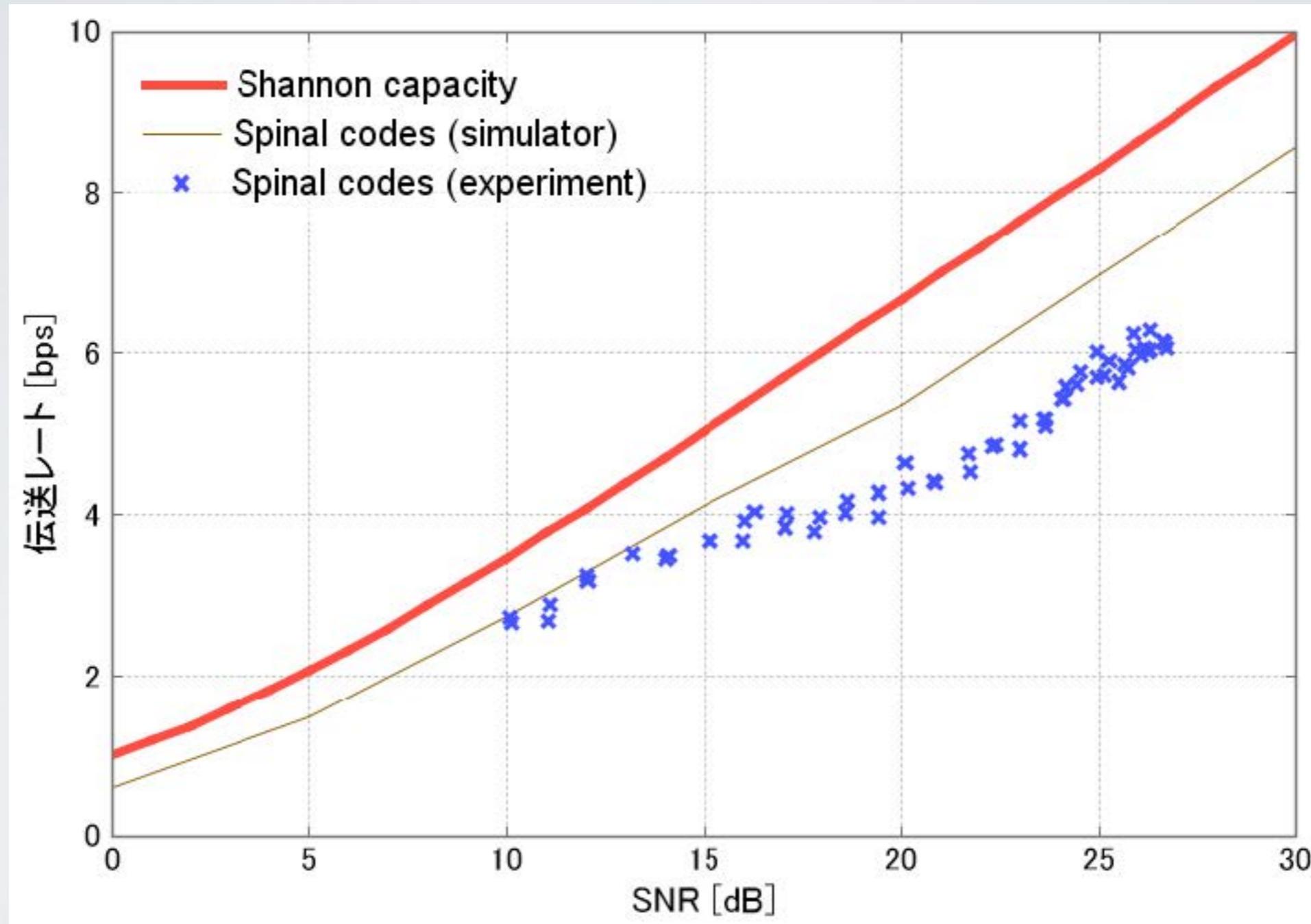


# 実測環境

- 物理層パラメータ
  - IEEE802.11aに準拠 (中心周波数5.11GHz、帯域幅2MHz)
  - 48サブキャリア (64-FFT)



# 実験結果

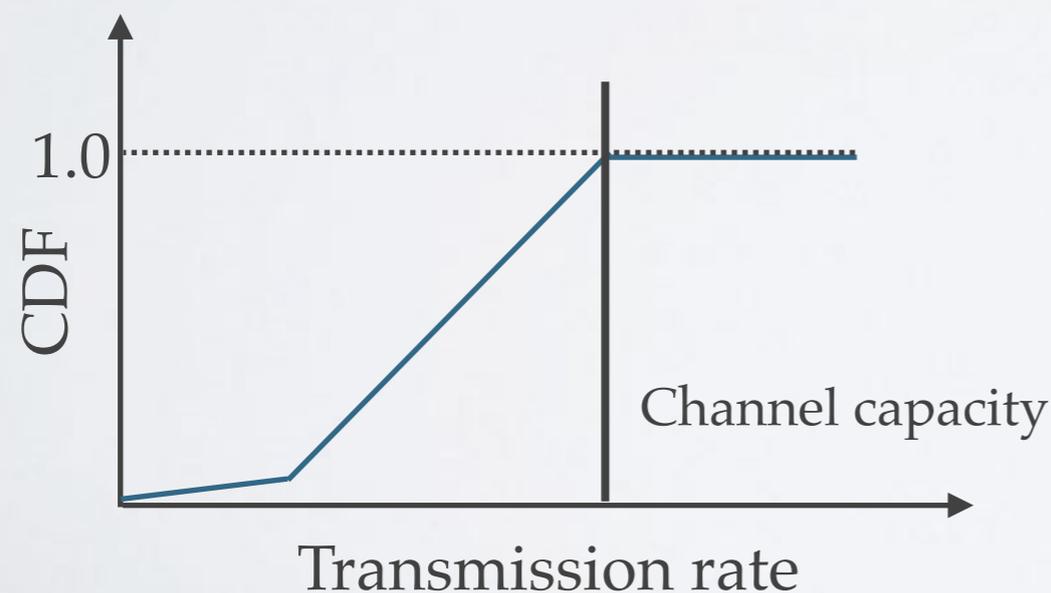


※ $n=256$ ,  $k=4$ , 4096QAM, Bubble Decoder ( $B=256$ ,  $d=1$ )

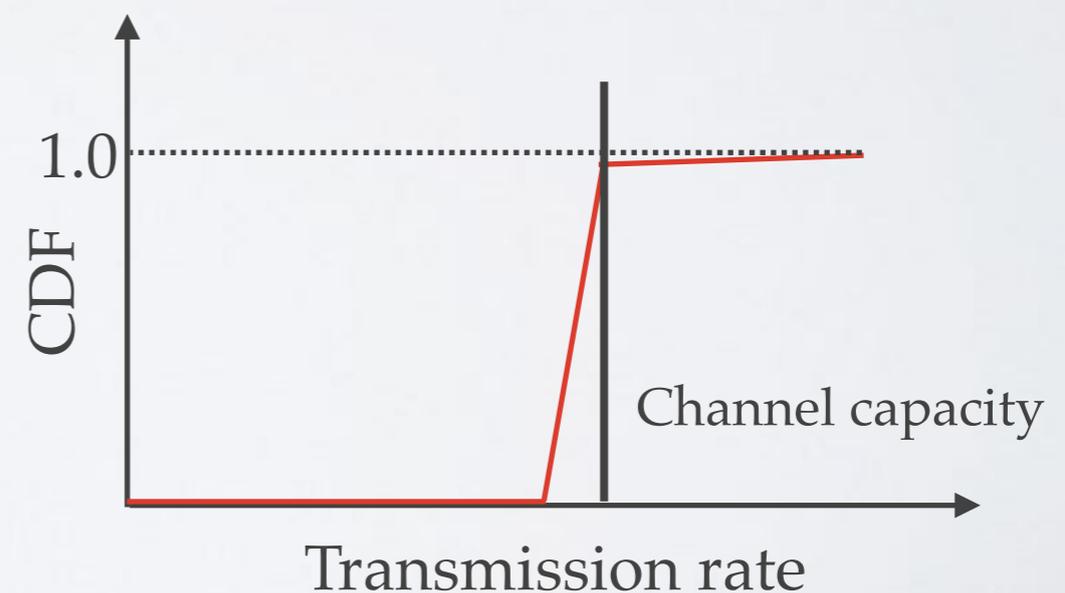
# Spinal符号は本当によい符号か？

- Perryらの論文と同様に、シミュレーションでも実環境でも高い平均データレート特性を示すことを確認。
- では伝送レートの累積密度分布(CDF)やビット誤り率特性(BER)はどうなっているのか？

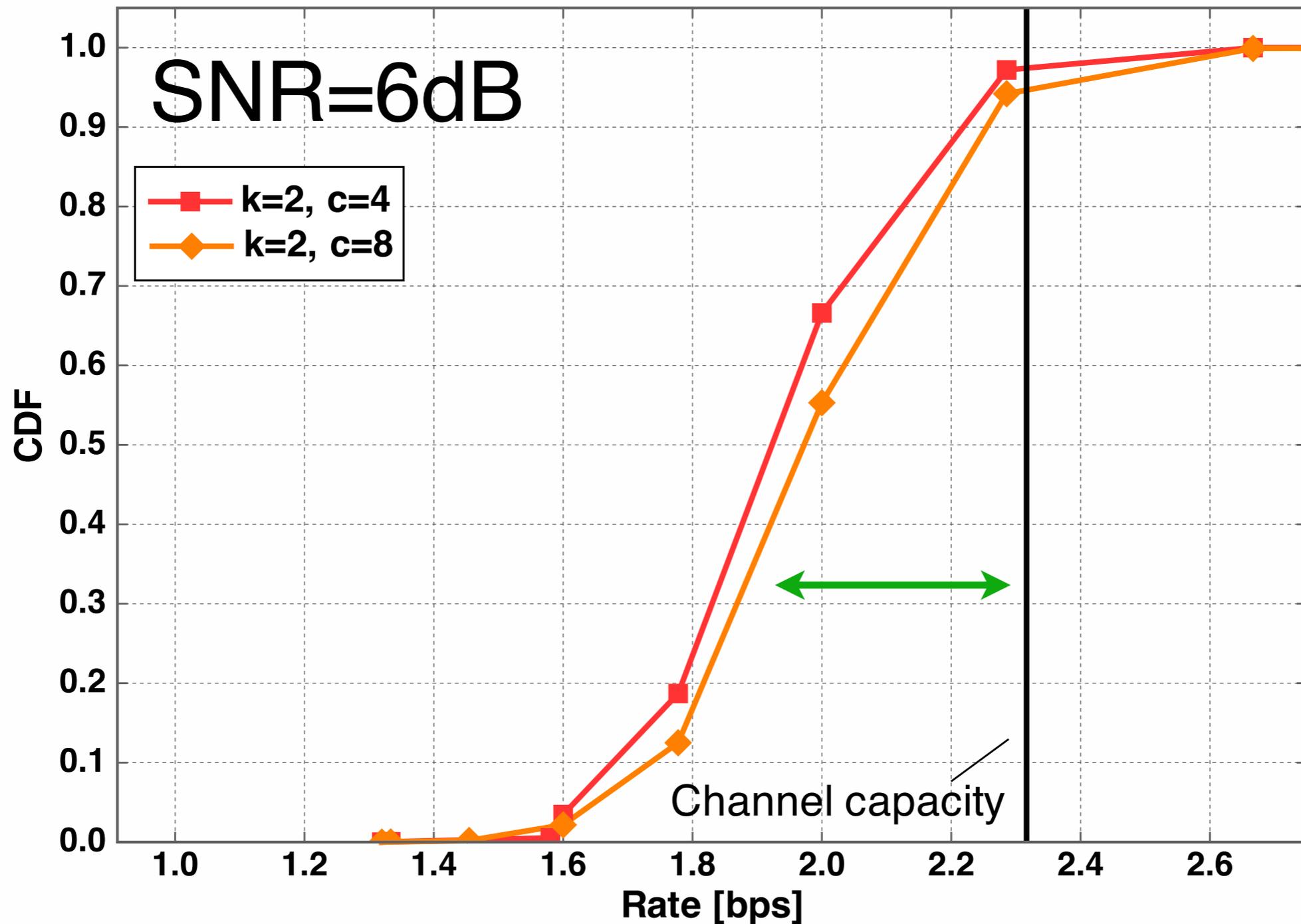
CDF example of a “bad” code



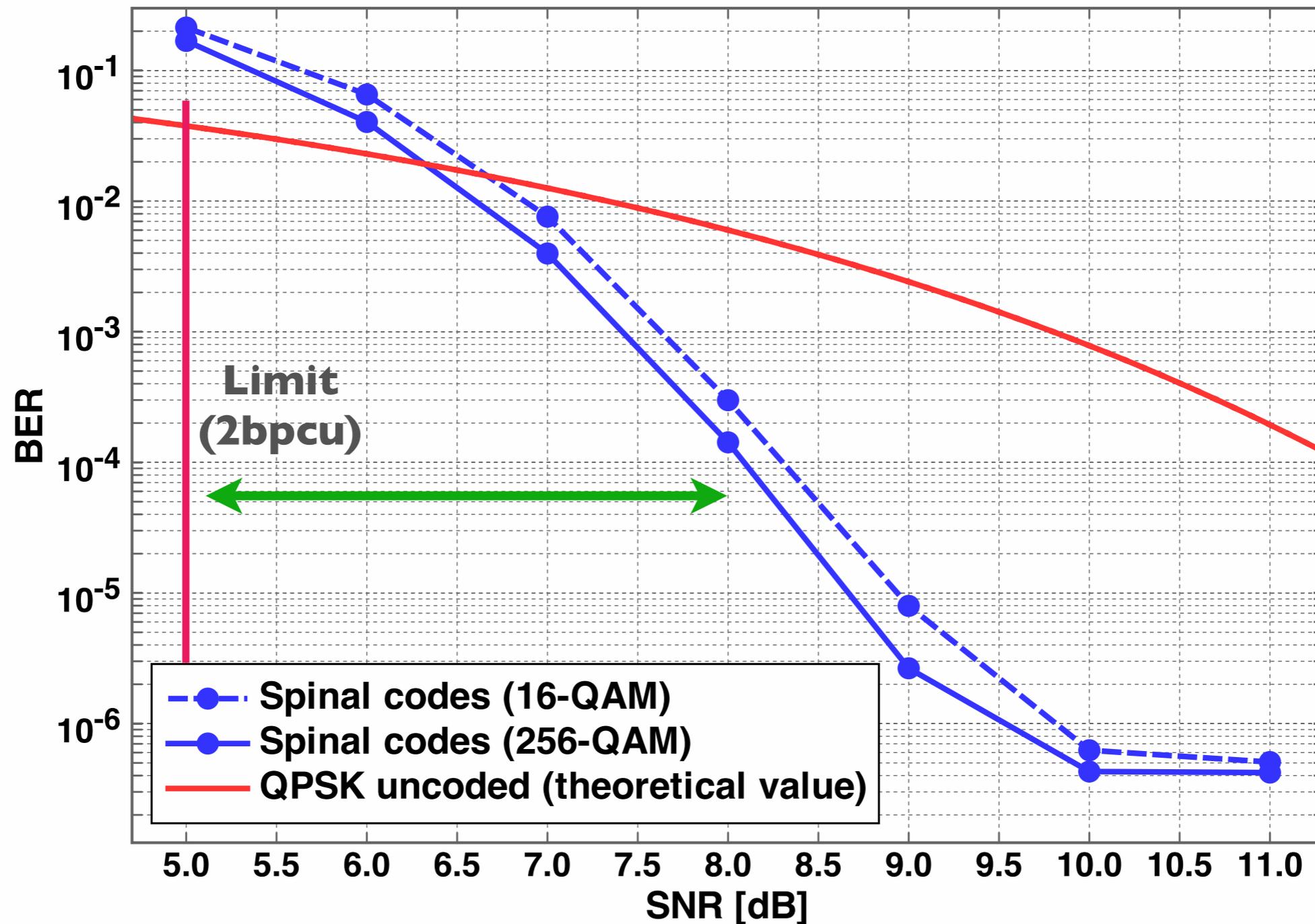
CDF example of a “good” code



# 伝送レートのCDF特性



# Spinal符号のBER特性



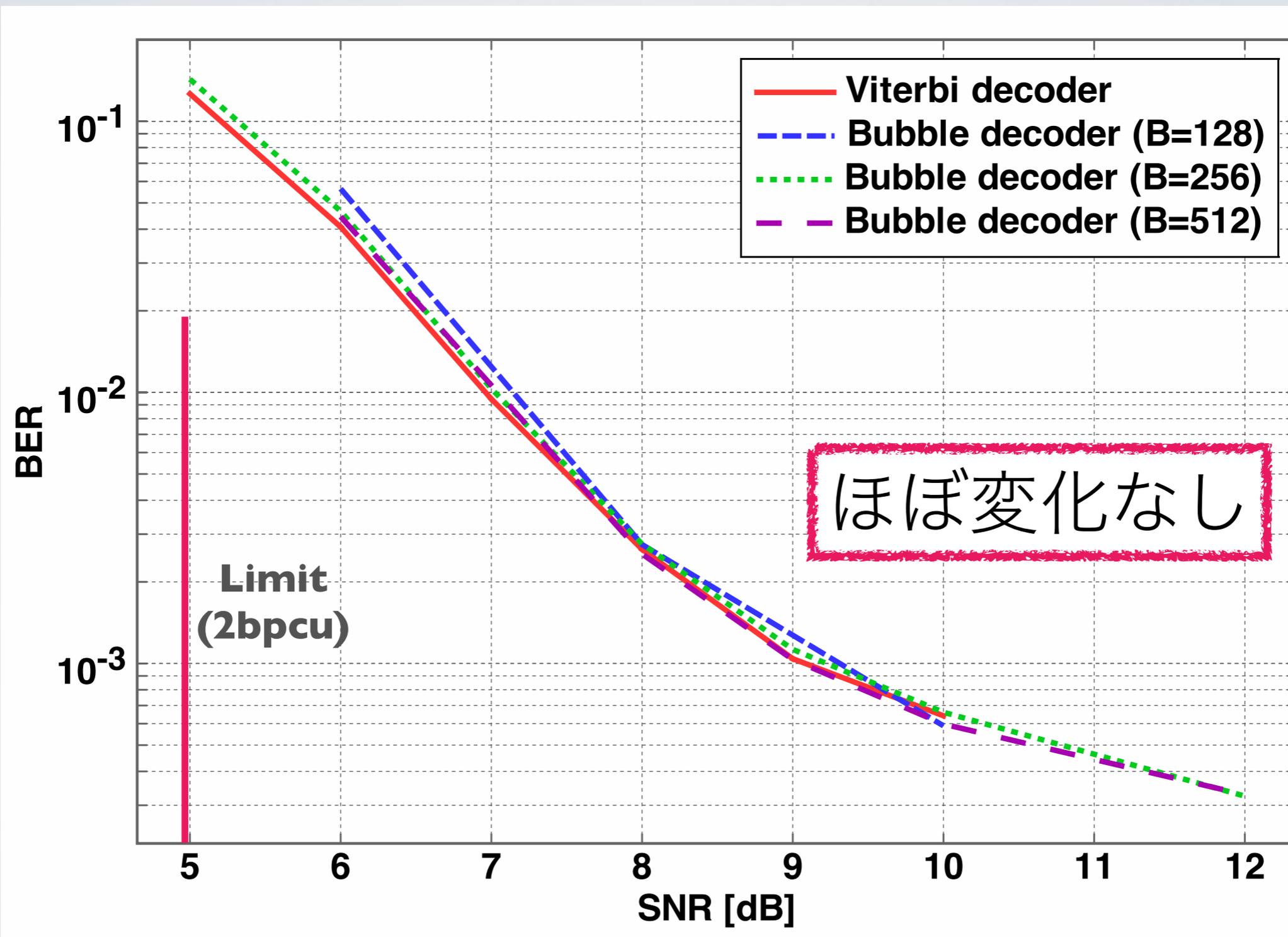
※ $n=128$ ,  $k=2$ ,  $v=32$ ,  $B=256$ ,  $d=1, 2$  [bpcu]

# 考察

- バブル復号はあくまでも最尤復号の近似アルゴリズム
- 正しいパスを途中で切り捨ててしまっている？
- 復号変えることで特性がどれだけ変わるかを確認
- 状態数を減らしてビタビ復号を実施

Data length	128
Data size $k$	2 [bits]
Modulation	4096QAM ( $c=12$ )
Hash size $v$	12
Decoder	Bubble, Viterbi decoder
Beam width	128, 256, 512
Depth	1

# ビタビ復号を用いたBER特性

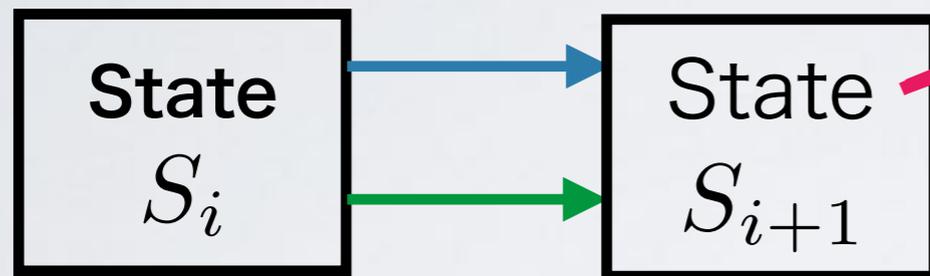


※ $n=128, k=2, v=12, d=1, 2$ [bpcu]

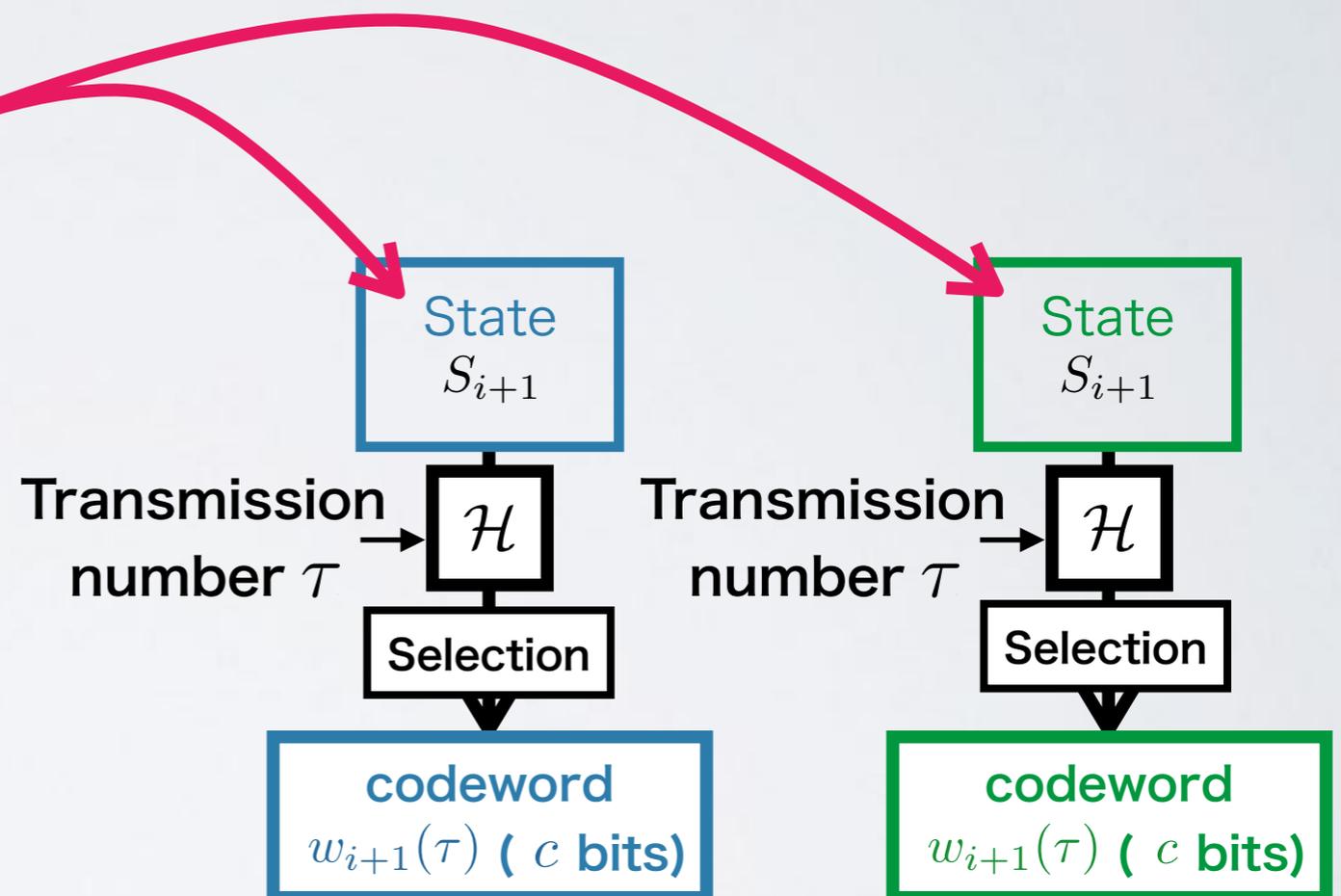
# Spinal符号の問題

## 状態の衝突：平行パス

情報サブブロック"0"



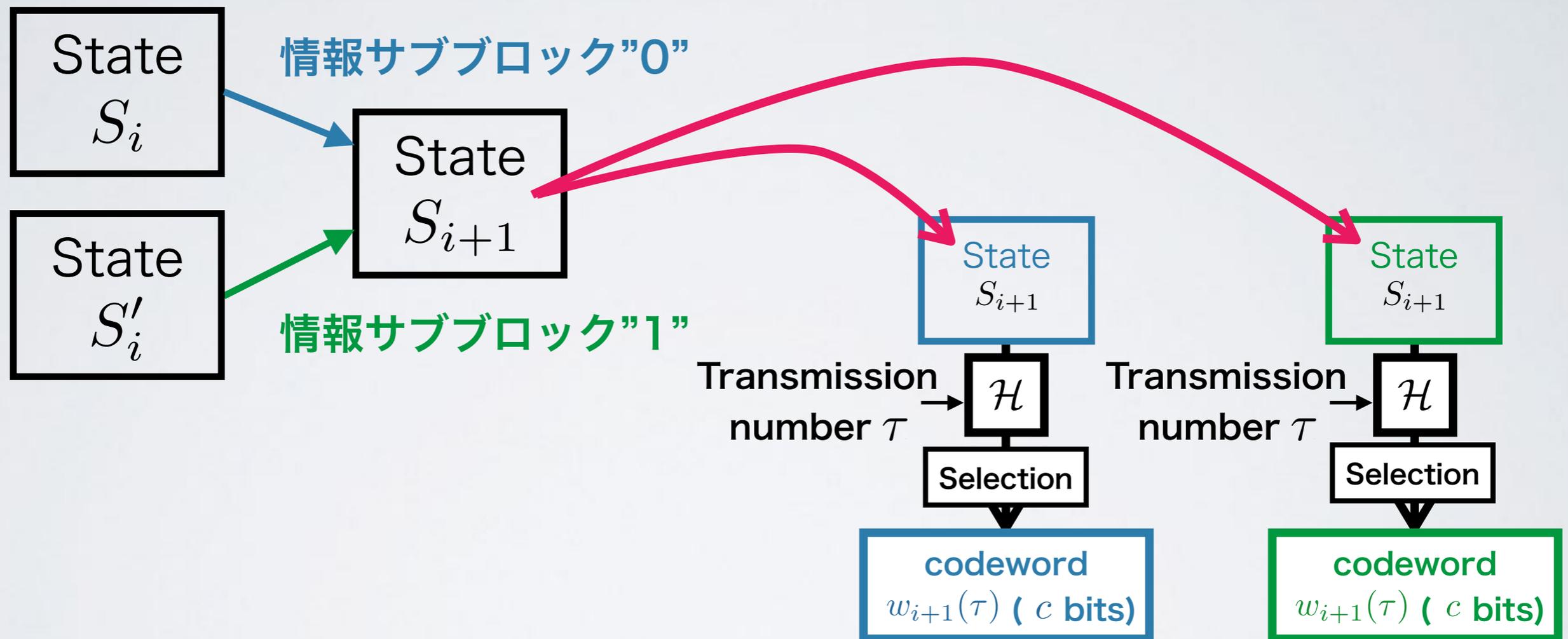
情報サブブロック"1"



常に同じ符号語

# Spinal符号の問題

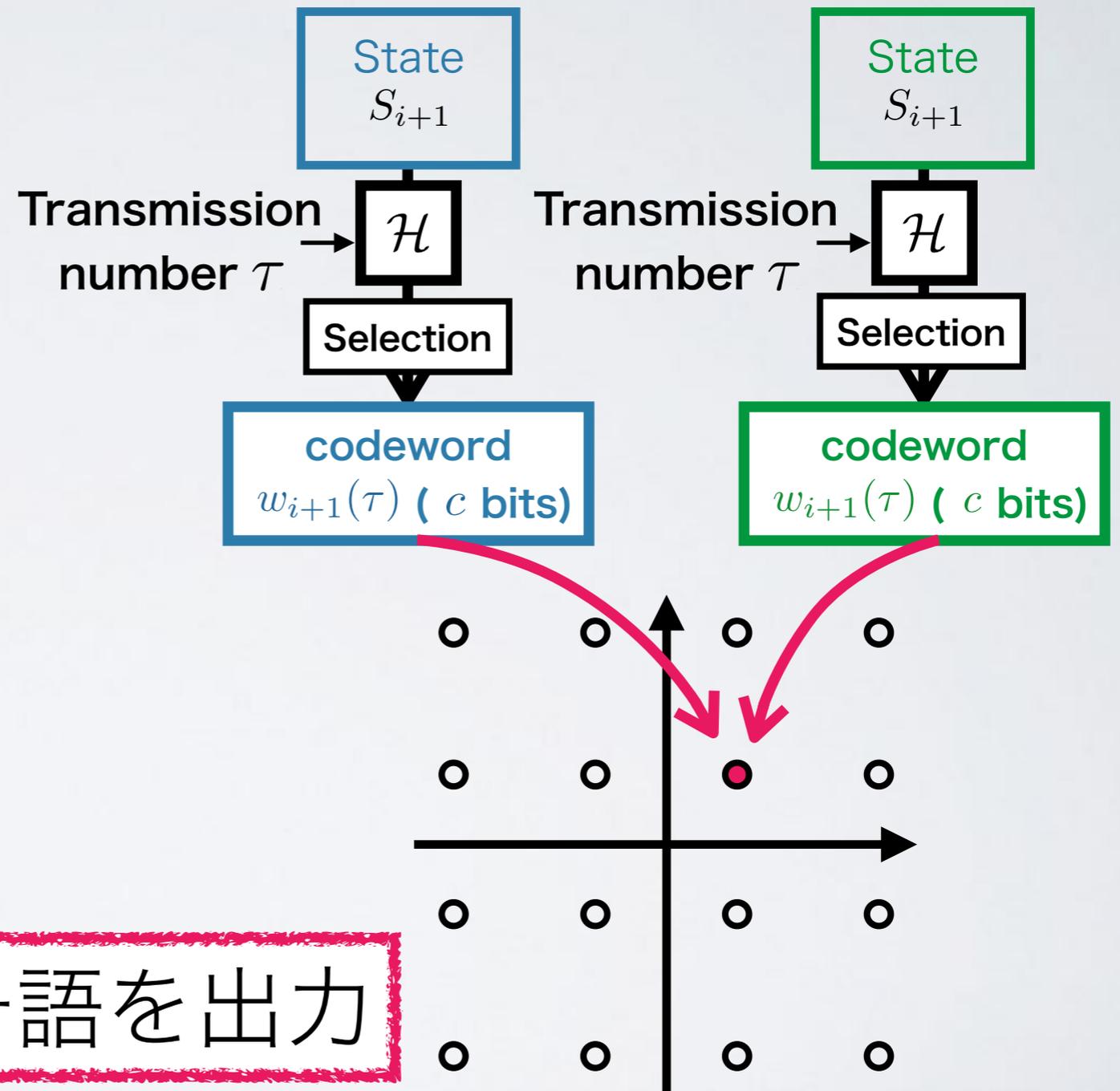
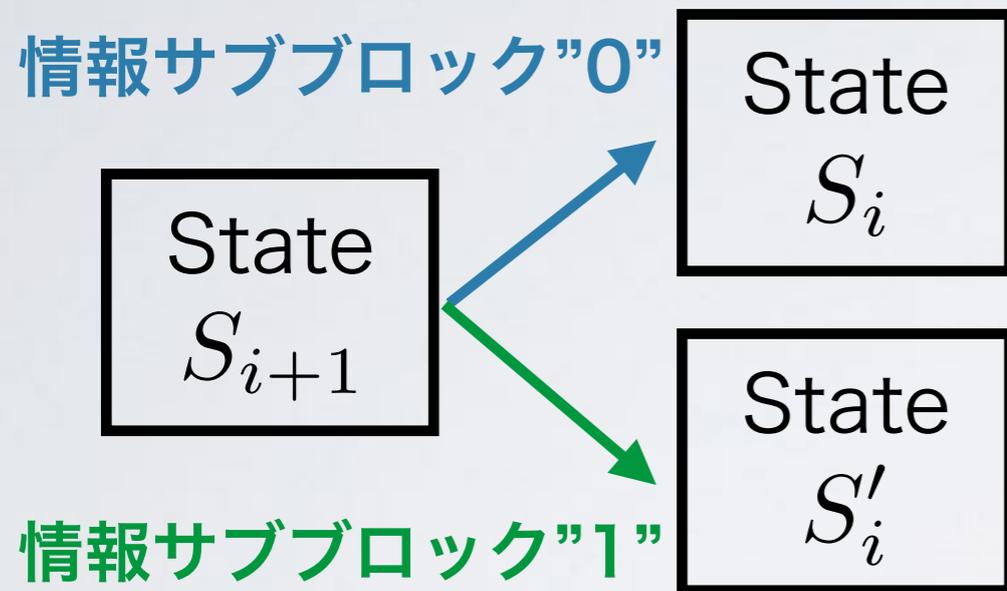
## 状態の衝突：パス合流



常に同じ符号語

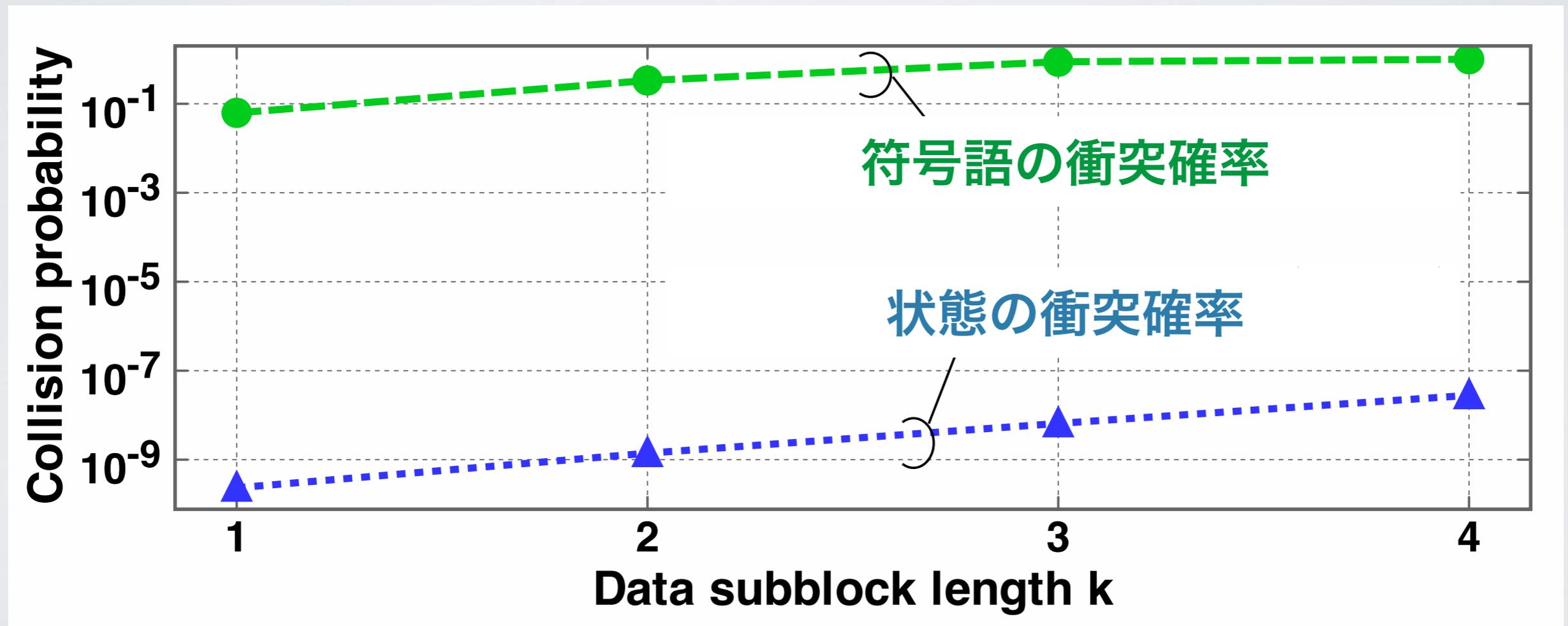
# Spinal符号の問題

## 符号語の衝突

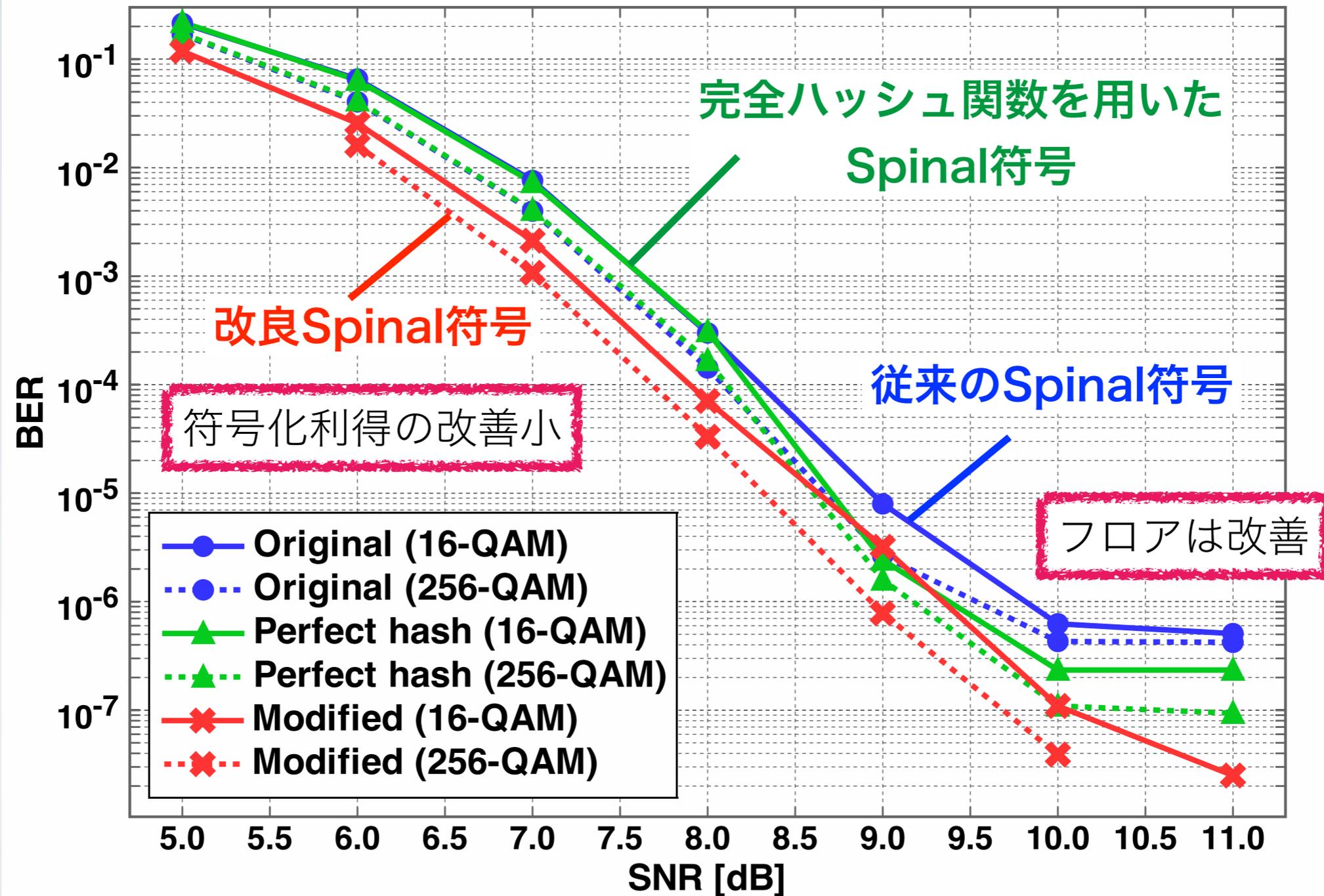


# 衝突確率

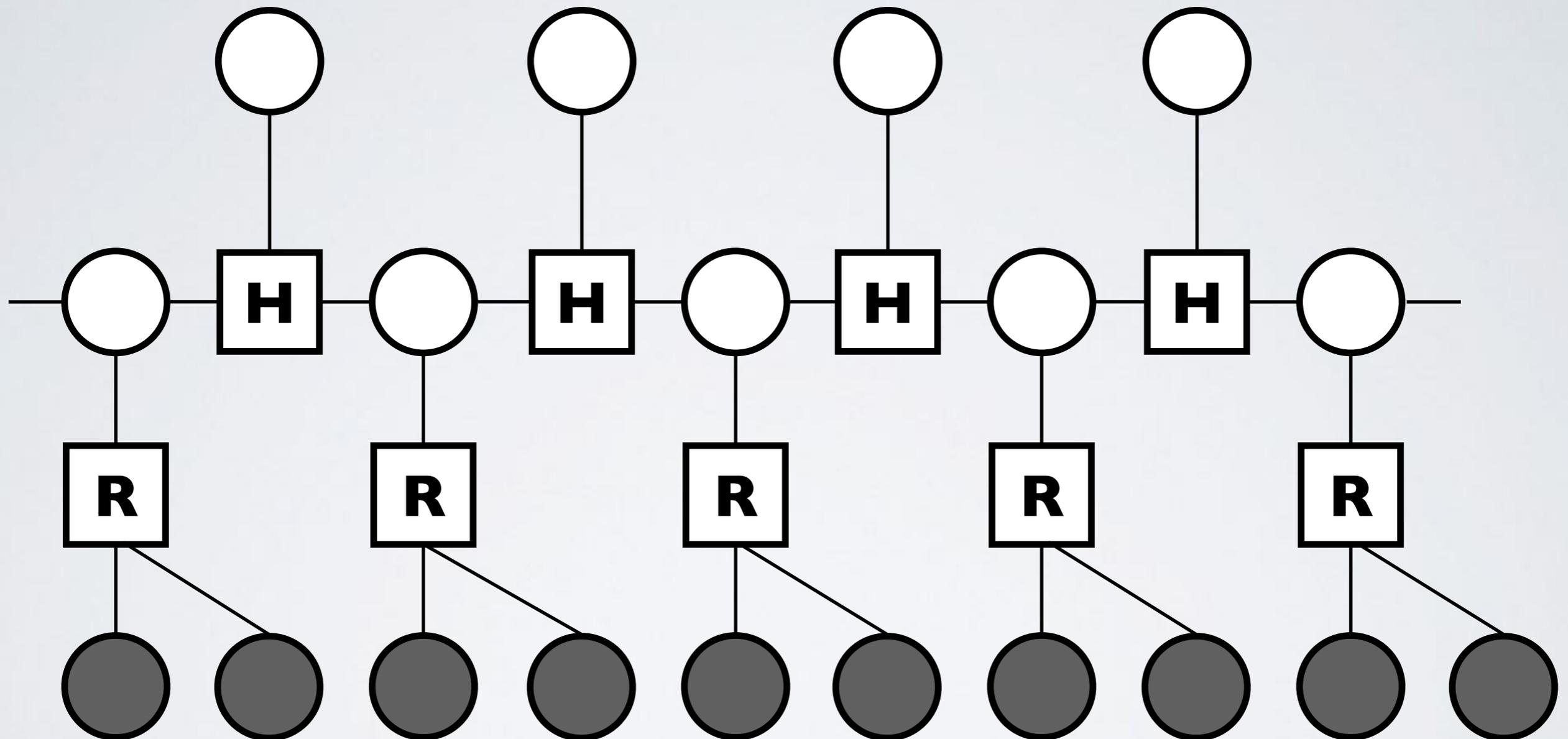
- 符号語長  $c = 4$ , 状態長  $\nu = 32$  であれば,  
符号語の衝突確率は状態の衝突確率のおよそ  $10^8$  倍



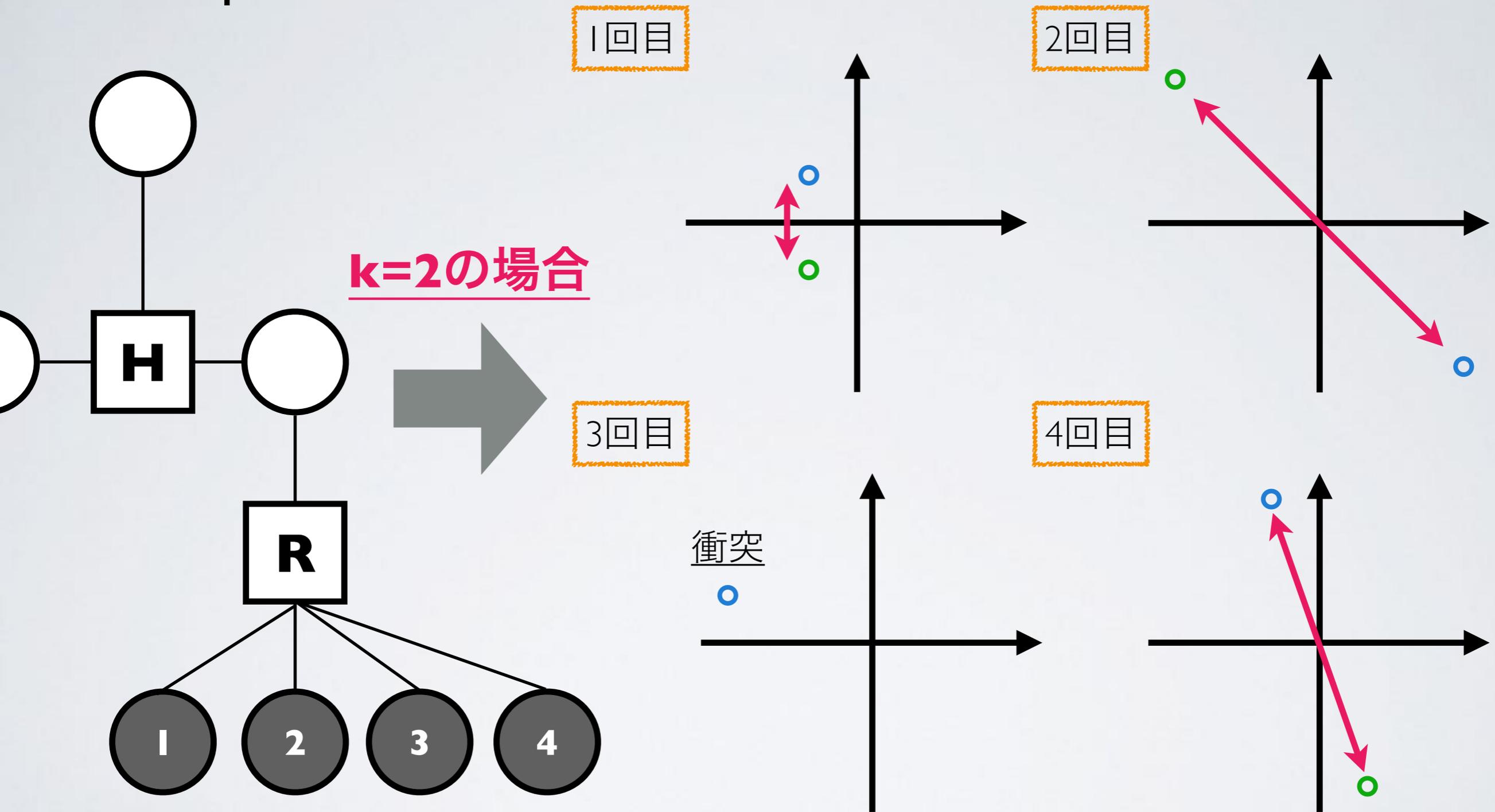
# 衝突の改善による効果



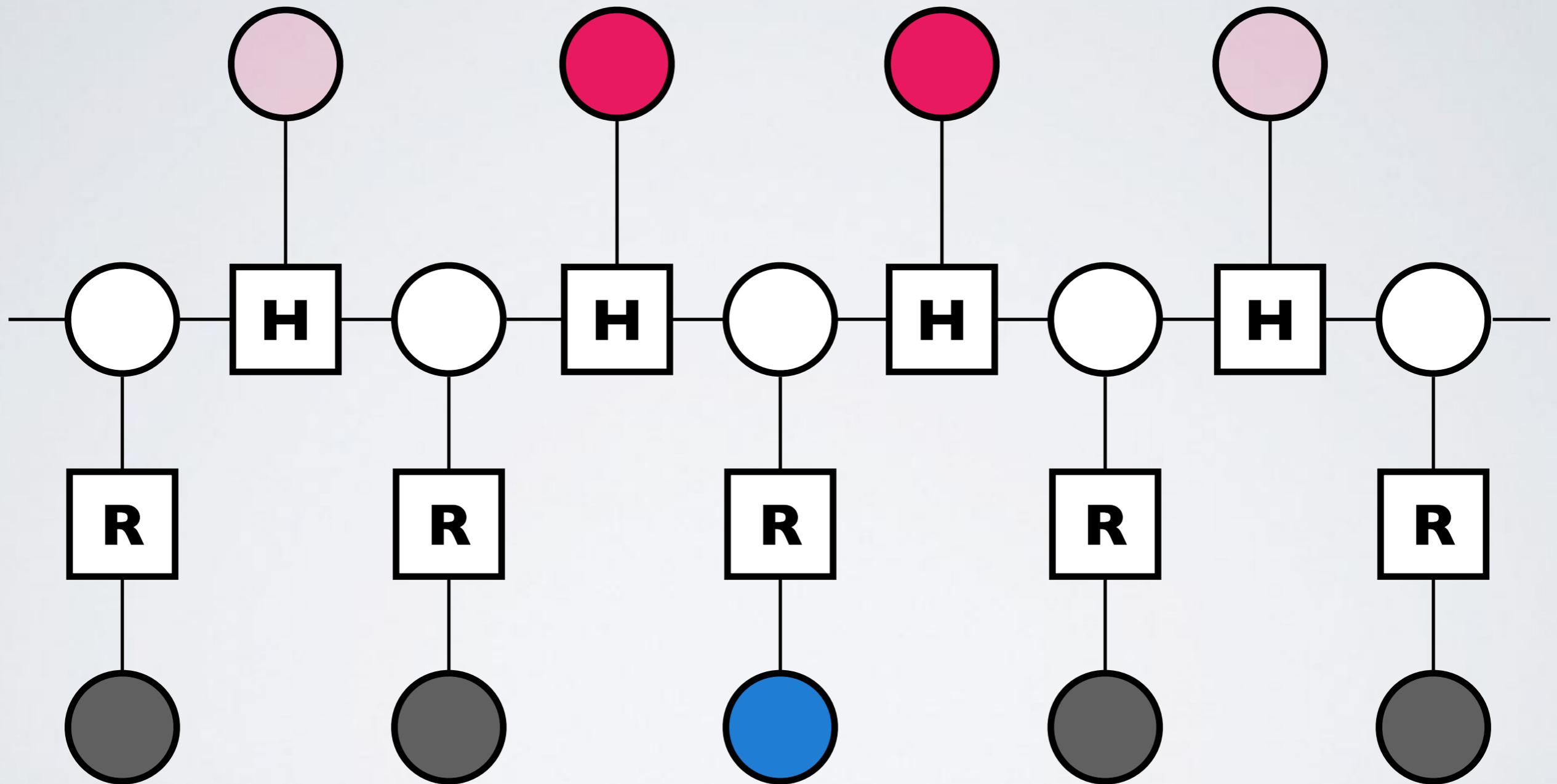
# Spinal符号のプロトグラフ表現



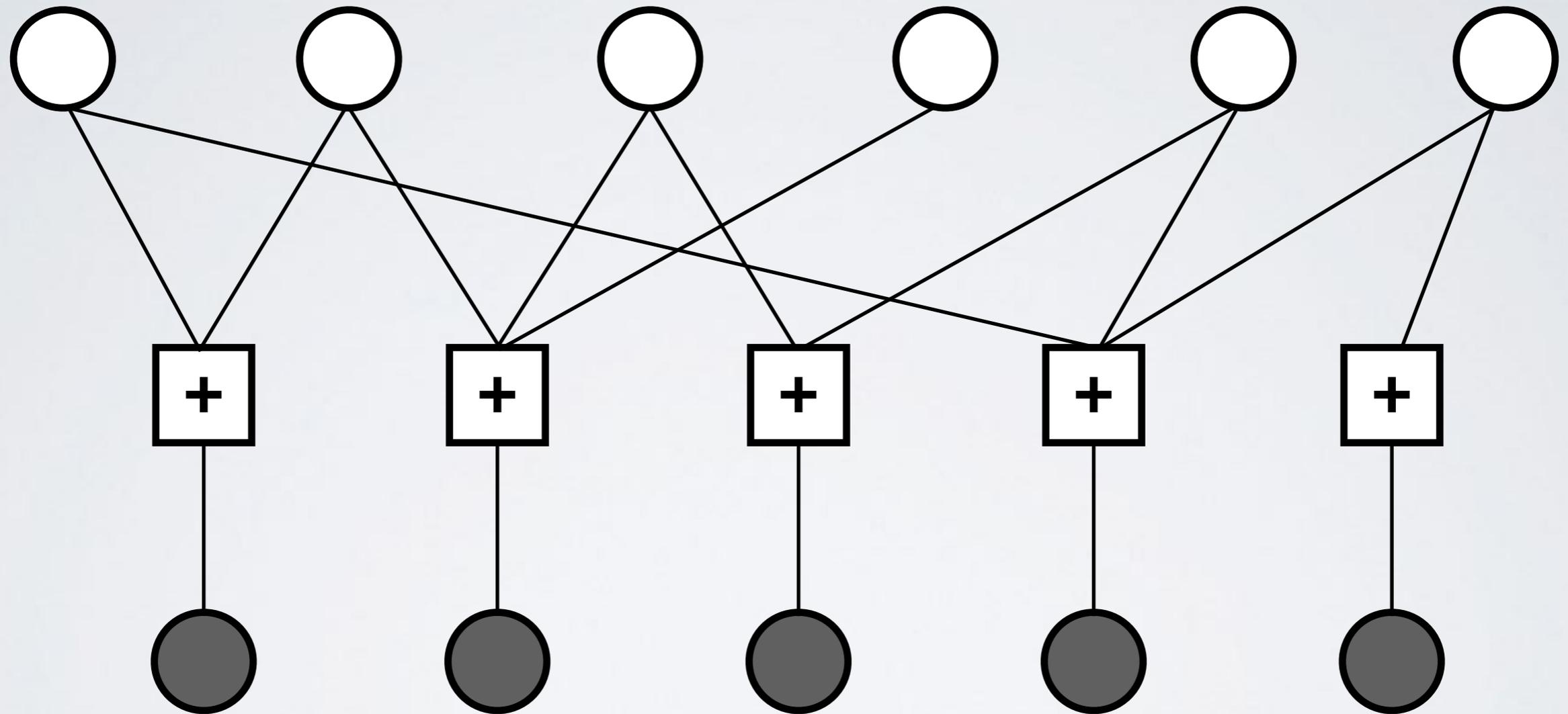
# Spinal符号における乱数性



# Spinal符号のプロトグラフ表現

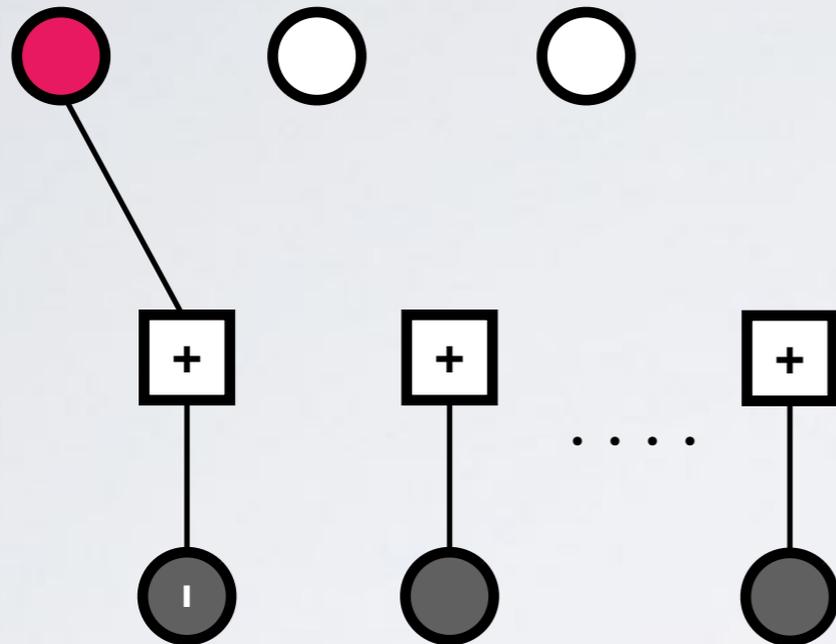


# LT符号のプロトグラフ表現

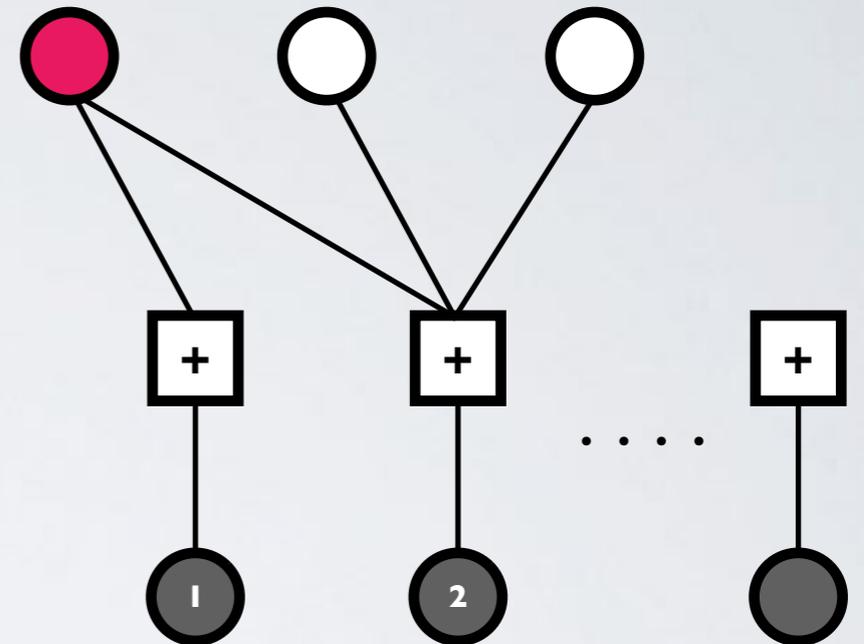


# LT符号の乱数性 (1/2)

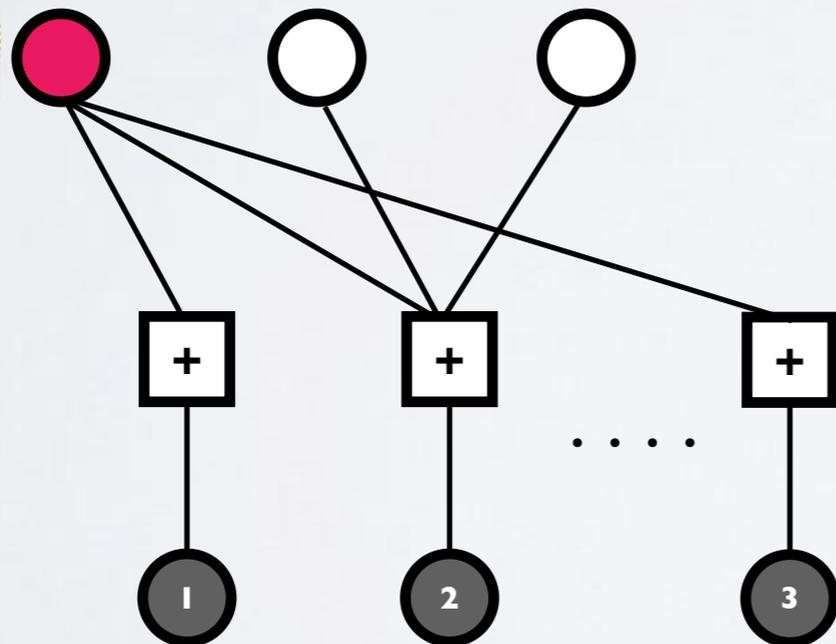
1回目



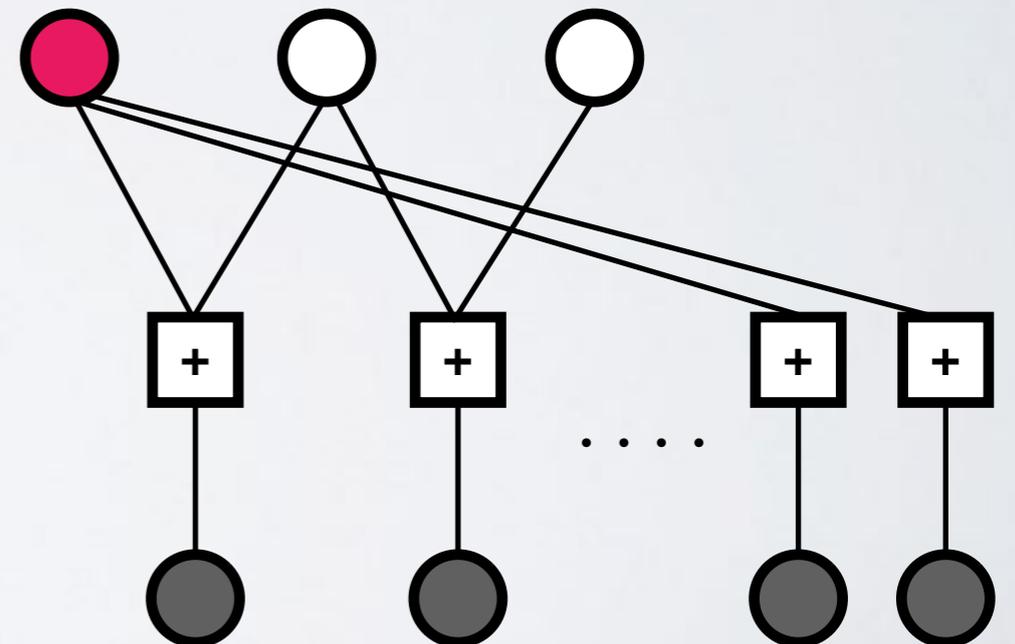
2回目



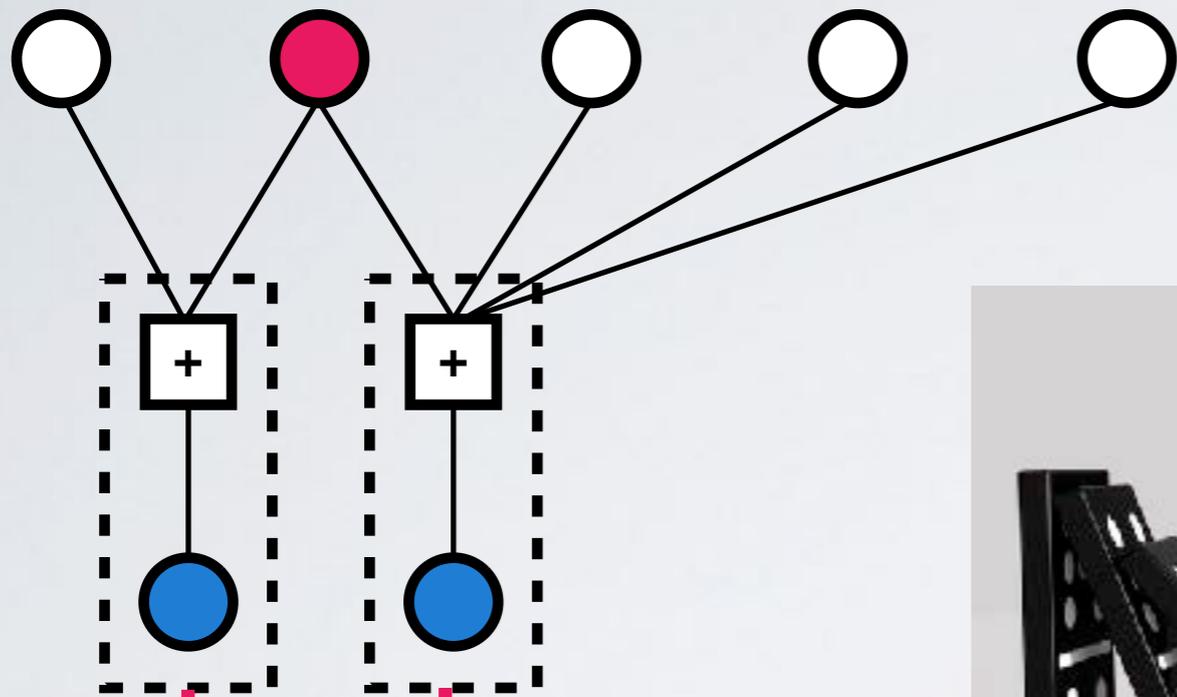
3回目



4回目



# LT符号の乱数性 (2/2)



XORされている情報が多 (密)

XORされている情報が少 (疎)

# 符号設計上の問題

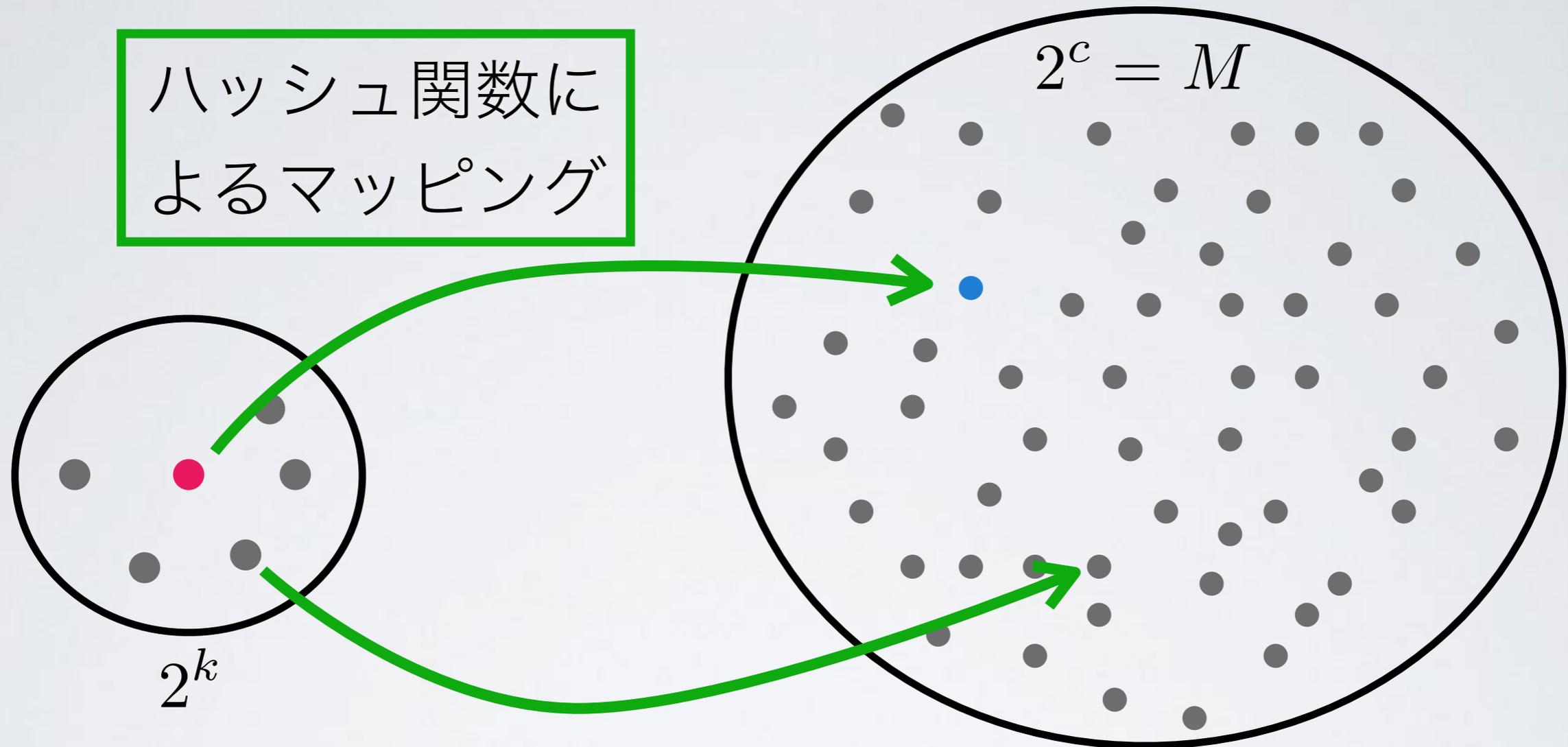
- つまり . . .

- Spinal符号では乱数性がある時点で出力される符号語のみに寄与
- LT符号では乱数性が異なる時点で出力される符号語からどれだけの情報を受け取るかという点に寄与

- Spinal符号の問題点

- 符号語生成が符号全体の特性改善にあまり寄与しない
  - 衝突を改善しても（ほぼ）木の「ある部分のみ」に貢献
- 膨大な状態数が符号生成に生かされていない
  - 各時点の符号語出力の最小ハミング距離は0（もしくは1）
- 疎密性(イレギュラー性)の不在
  - 確率伝播復号を使用しても特性の改善は見込めない

# 極論すると



Mが無限大になれば誤り率は0にできる

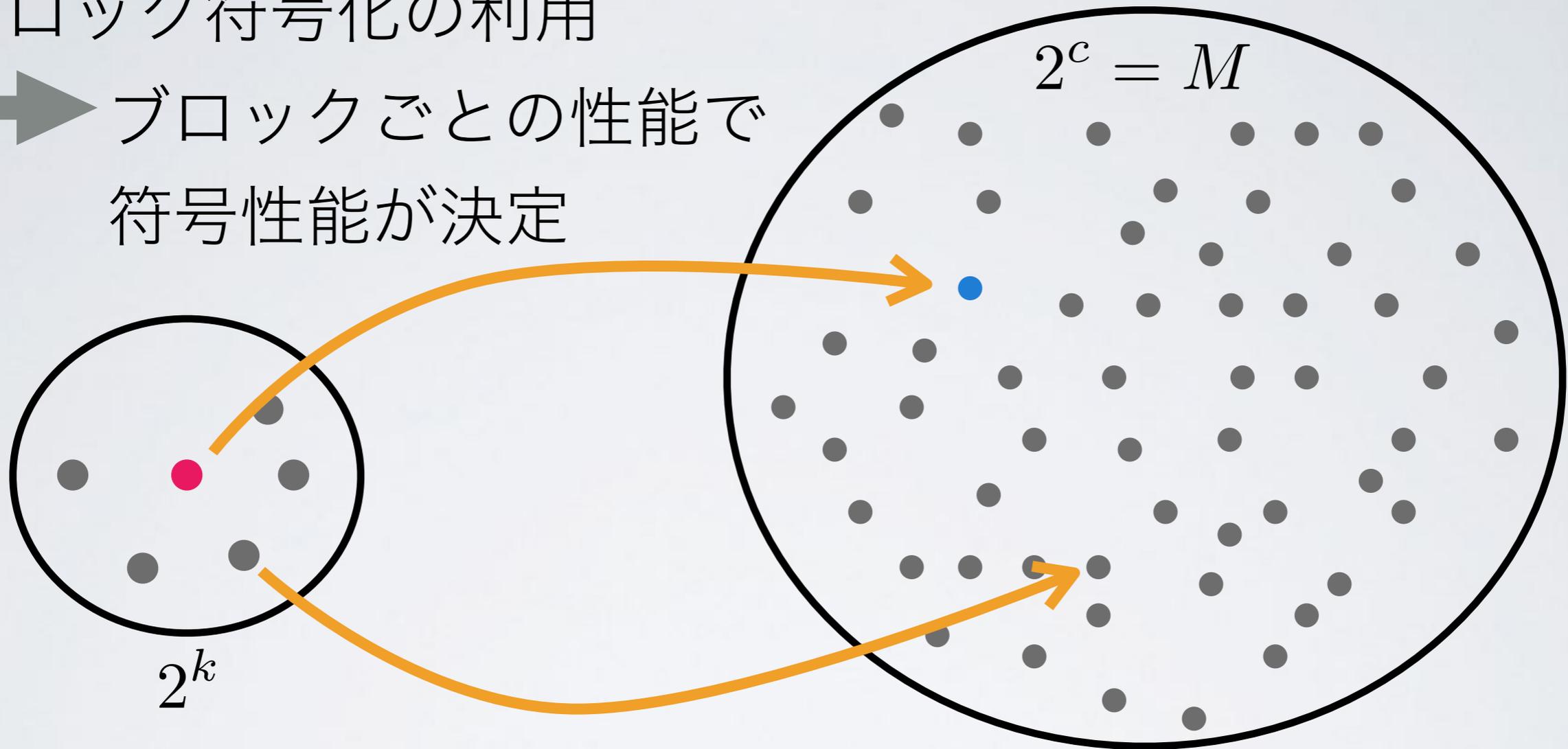


効率の悪い符号構成法

# 改善法

## 1. ブロック符号化の利用

➡ ブロックごとの性能で  
符号性能が決定



組織符号化の導入

➡ 最小距離は"1"

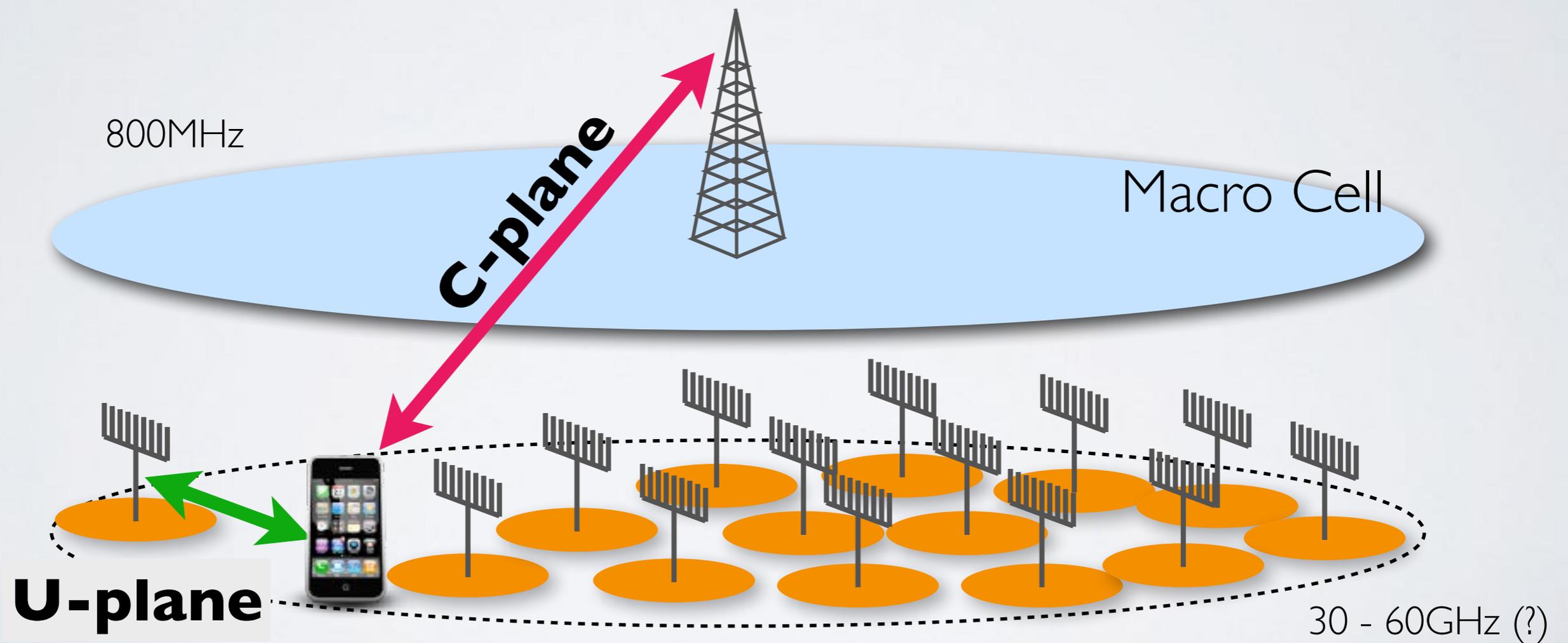
# Spinal符号の可能性

- Spinal符号が性能を発揮しうる領域：
  - $2^{32}$ -ary QAMなど**超多値変調**を用いた場合
- 現実的な仮定か？
  - 復調・復号に十分な位相同期がとれない
  - A/D, D/Aのビット数が足りず復調できない
  - PAPRが高く、増幅器による非線形歪みの影響大
- 現実的な解は構造を変える
  - ハッシュ関数をやめて構造的にする
  - 乱数性を活かす構造を導入する

例：超低レート畳み込み符号
- **LT符号(Raptor符号)に対して利点がありません・・・。**

# 期待されるレートレス符号

- 第5世代移動体通信(5G)以降のキーワード
  - C/U分離：ファントムセル、ブースターセル等



# 期待されるレートレス符号



レートレス符号によってフィードバックなどを用いることなく、常に通信路容量に漸近するレートで通信

# 期待されるレートレス符号

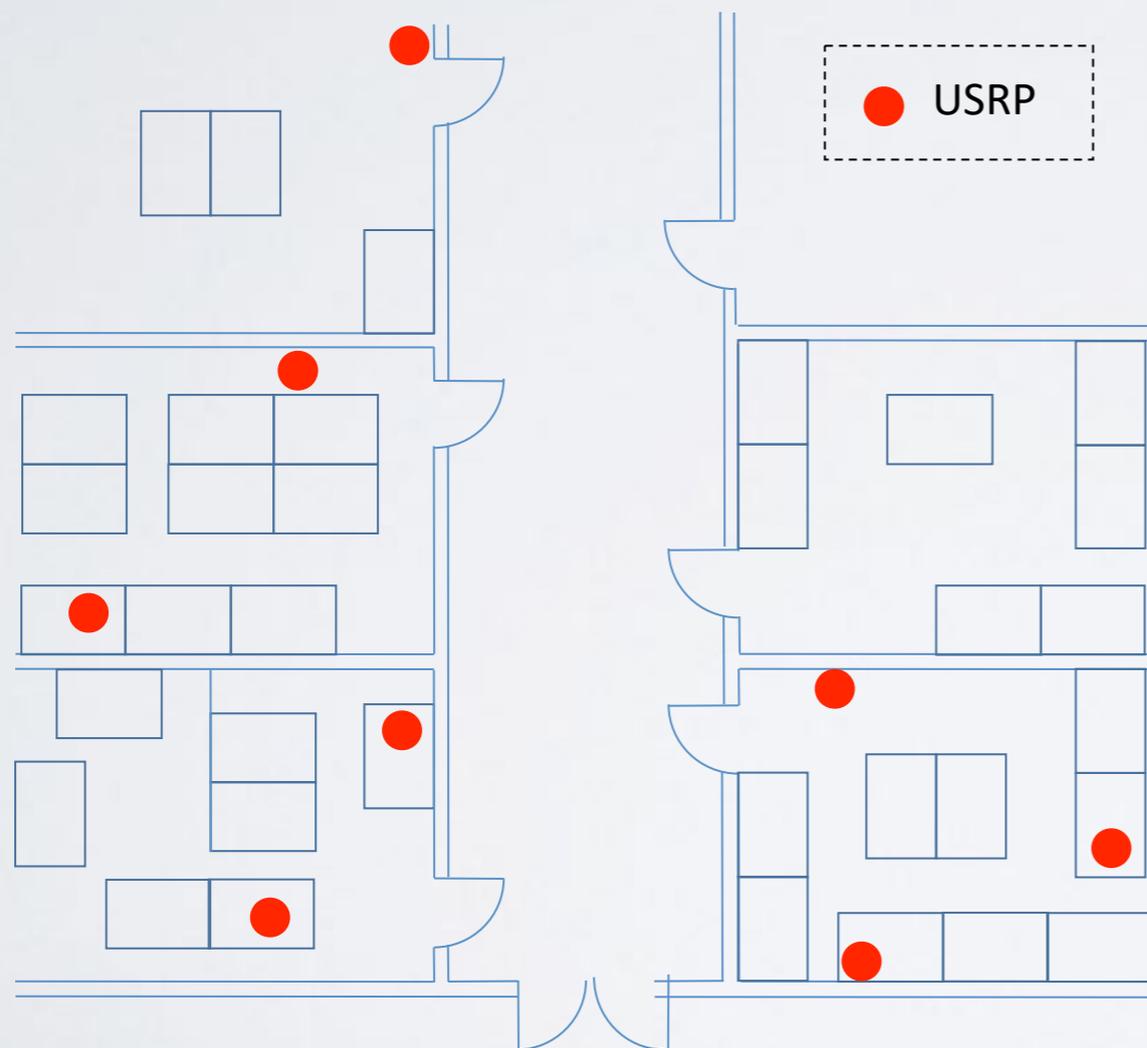


レートレス符号と重畳通信の組み合わせによってシステムスループットの向上

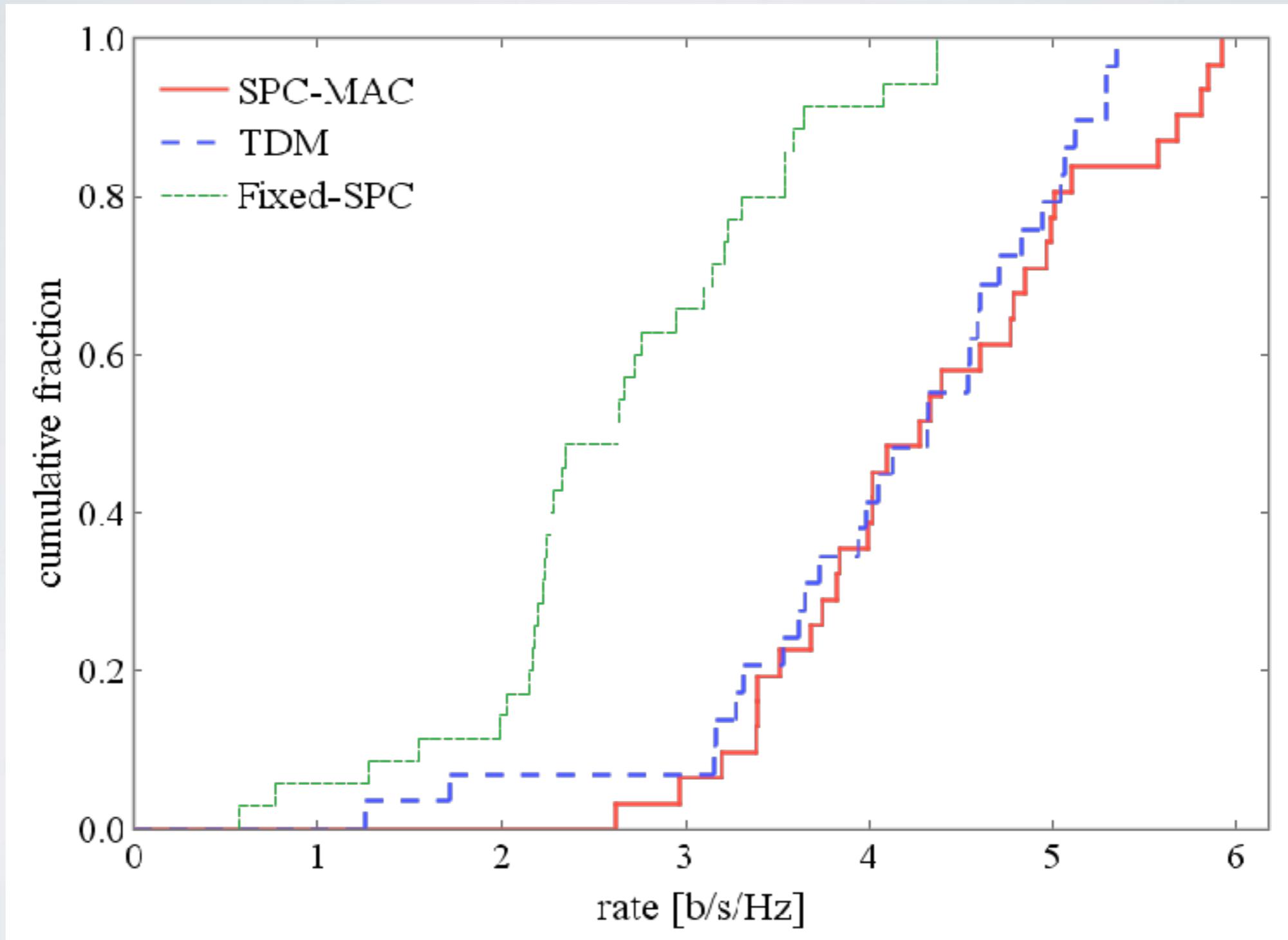
# Spinal符号を用いた重畳符号化MAC

[1] 西田 昇平, 山崎 景太, 石橋 功至, 猿渡 俊介, 渡辺 尚, "Spinal符号を用いた重畳符号化伝送の基礎検討", 電子情報通信学会総合大会, 2014年03月.

[2] 山崎 景太, 西田 昇平, 石橋 功至, 猿渡 俊介, 渡辺 尚 "ブロードキャスト通信路容量を達成するための無線通信方式の実装について" 情報処理学会研究報告, モバイルコンピューティングとユビキタス通信研究会, MBL-70-1, March 2014., 2014年03月.



# 実験結果



# まとめ

- Spinal符号の基礎、実装、解析について
- Spinal符号の問題点を明らかに
  - 今後：符号構造上の問題を理解した上でどういう符号を作るか？
- CS系の学会では符号や変調を扱った論文は多いが、問題が正しく理解されていない
  - 符号屋さん、通信理論屋さんの出番
- 次世代の無線通信技術における新しい符号の可能性
  - 5Gにおけるレートレス符号
  - レートレス符号化による重畳符号化

# 謝辞

- 本研究の遂行にあたり、以下の皆様に多くのご助言、ご協力を頂きました。心より御礼申し上げます。
  - 大阪大学 渡辺 尚 先生
  - 静岡大学 猿渡 俊介 先生、山崎 景太 君、西田 昇平 君
  - 電気通信大学 尾形 駿 君、武石 直樹 君
  - 大阪大学 衣斐 信介 先生
  - 東京農工大学 杉浦 慎哉 先生
  - 香川大学 石井 光治 先生