Point-to-point codes for interference channels: A journey toward high performance at low complexity

Young-Han Kim

University of California, San Diego

Communication Theory Workshop
Dana Point, California
Tuesday, May 12, 2015

Point-to-point codes for interference channels: A journey toward high performance at low complexity

Young-Han Kim

University of California, San Diego

Communication Theory Workshop
Dana Point, California
Tuesday, May 12, 2015

Seok-Ki Ahn, Bernd Bandemer, Chiao-Yi Chen,
Abbas El Gamal, Kwang Taik Kim, Hosung Park,
Eren Şaşoğlu, and Lele Wang

Point-to-point codes for interference channels: A journey toward high performance at low complexity

Young-Han Kim

University of California, San Diego

Communication Theory Workshop
Dana Point, California
Tuesday, May 12, 2015

Seok-Ki Ahn, Bernd Bandemer, Chiao-Yi Chen, Abbas El Gamal, Kwang Taik Kim, Hosung Park, Eren Şaşoğlu, and Lele Wang

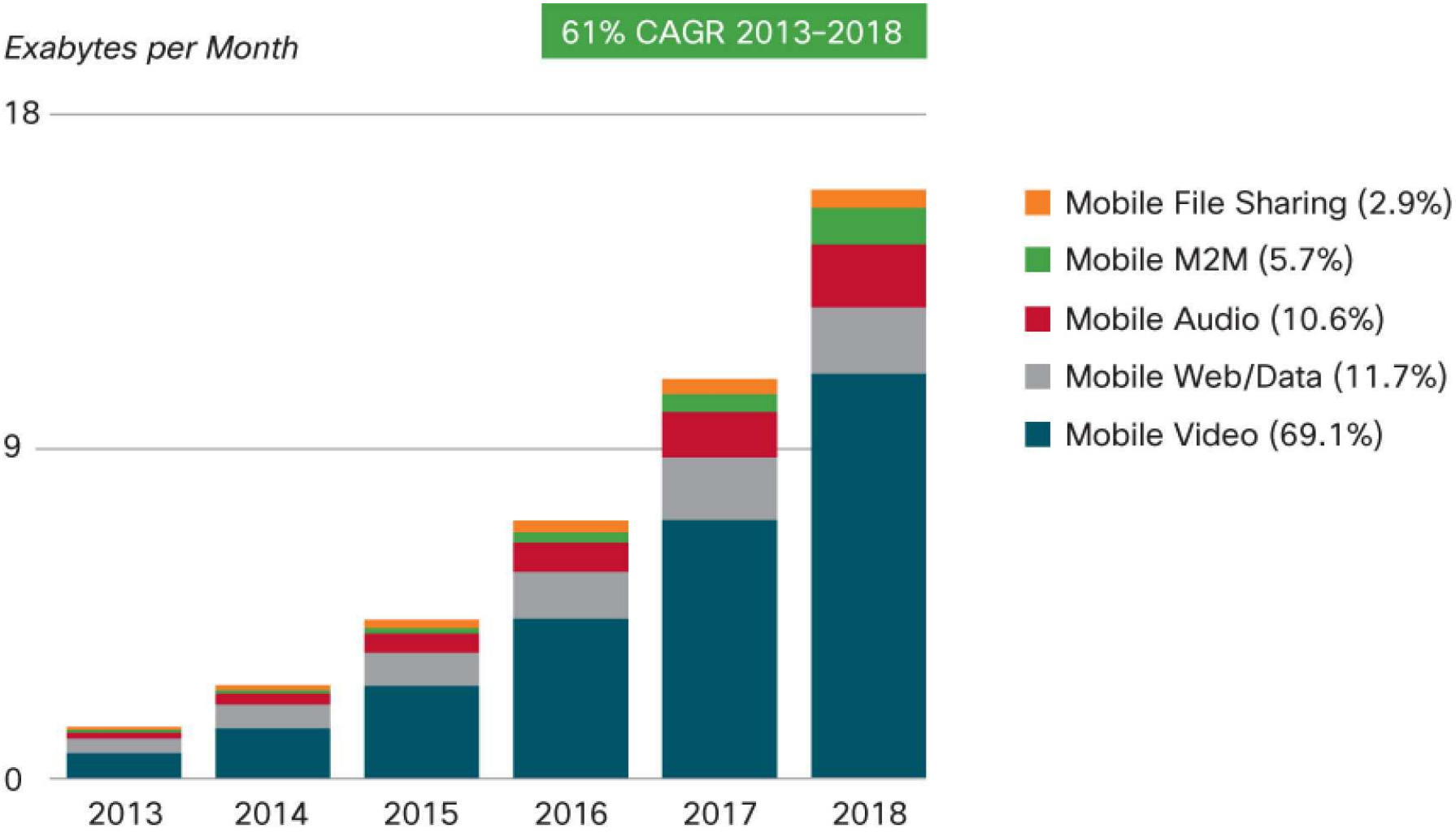
SPECIAL
Report

WORDS **BENNETT RING** ILLUSTRATION **NIGEL BUCHANAN**

CONNECTIVITY
~IS KING~

From smart watches that synchronise with smartphones, to portable high-definition cameras that can be remotely monitored from anywhere on Earth, 2014 has been a year in which to be linked in is everything.

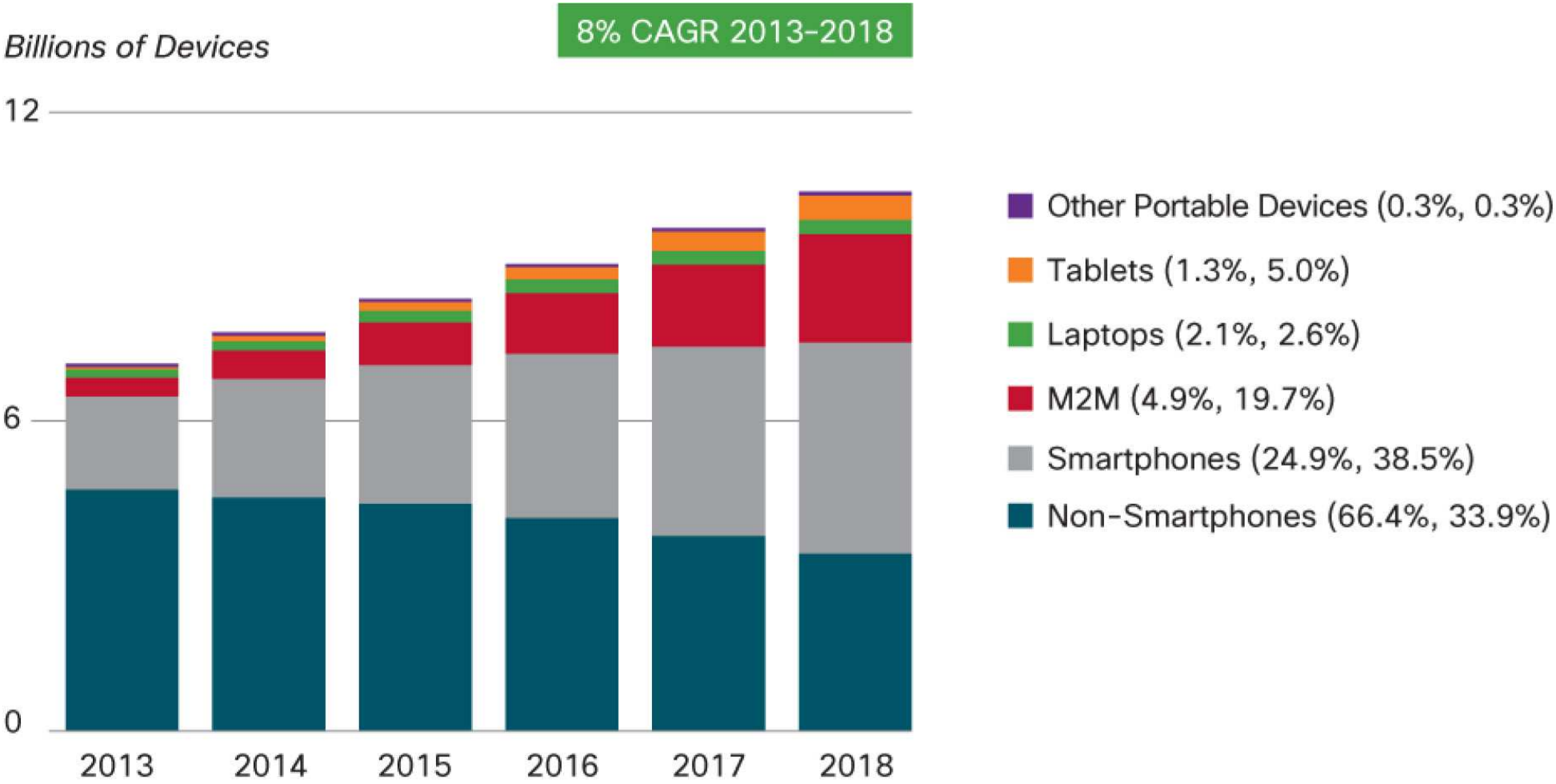
Where is wireless going?



Figures in parentheses refer to traffic share in 2018.

Source: Cisco VNI Mobile, 2014

Where is wireless going?

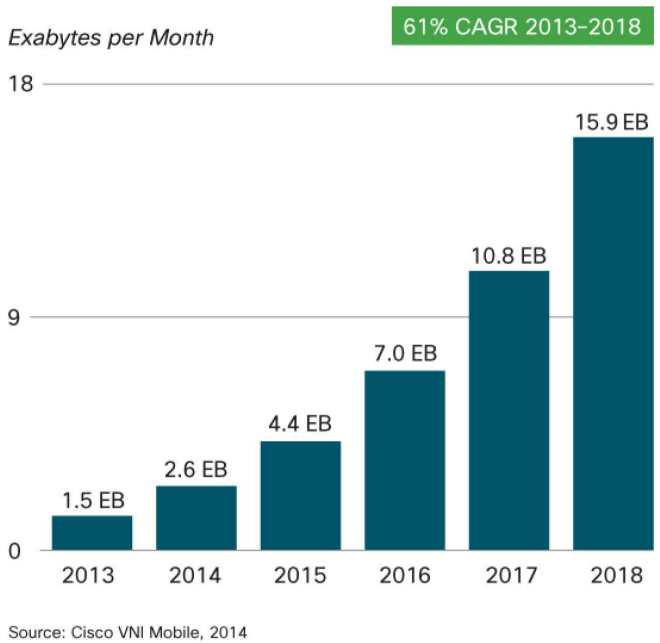


Figures in parentheses refer to device or connections share in 2013, 2018.

Source: Cisco VNI Mobile, 2014

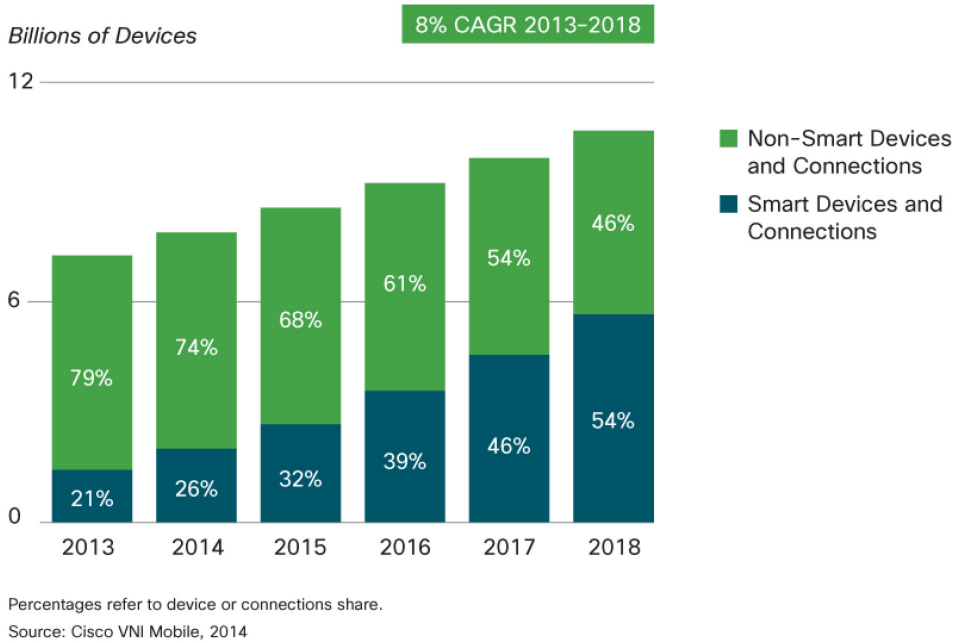
Where is wireless going?

Mobile data per month



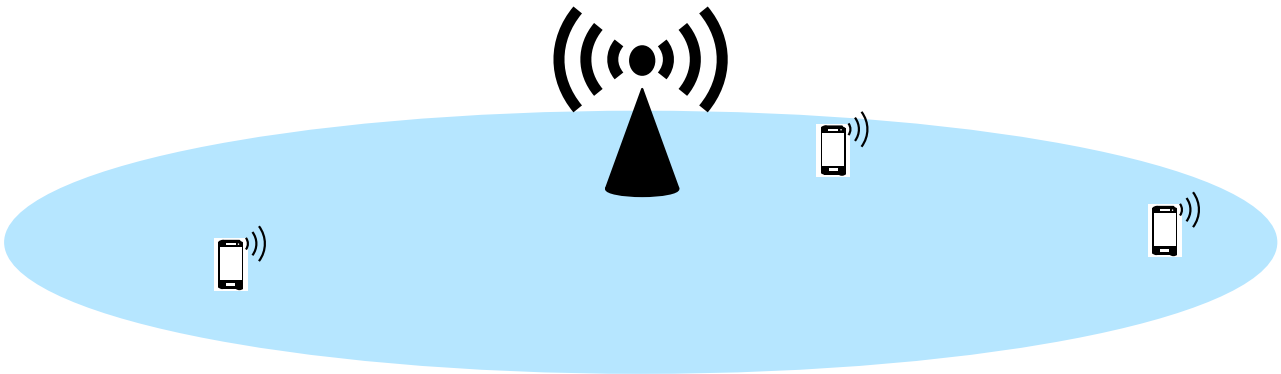
Source: Cisco VNI Mobile, 2014

Number of devices



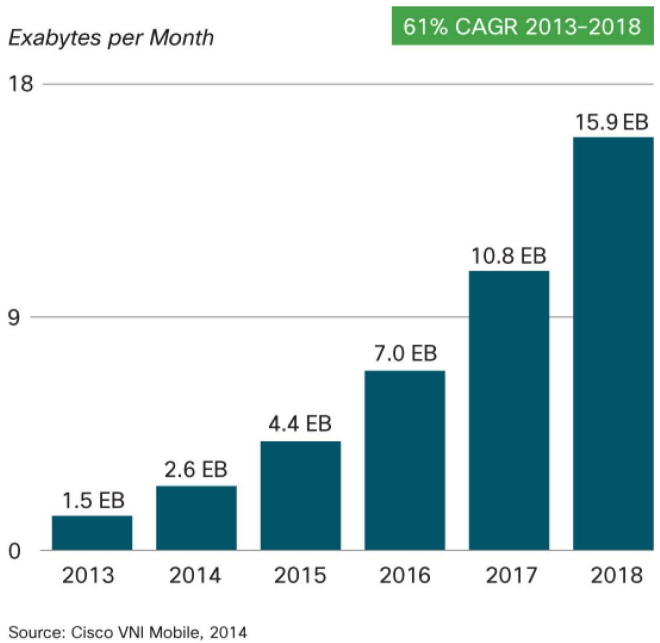
Percentages refer to device or connections share.

Source: Cisco VNI Mobile, 2014



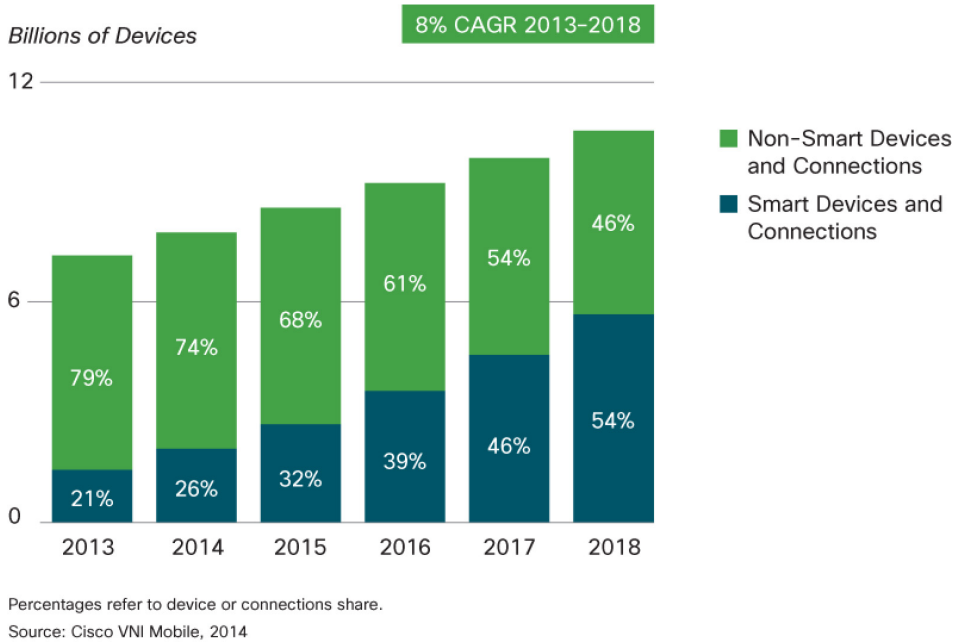
Where is wireless going?

Mobile data per month



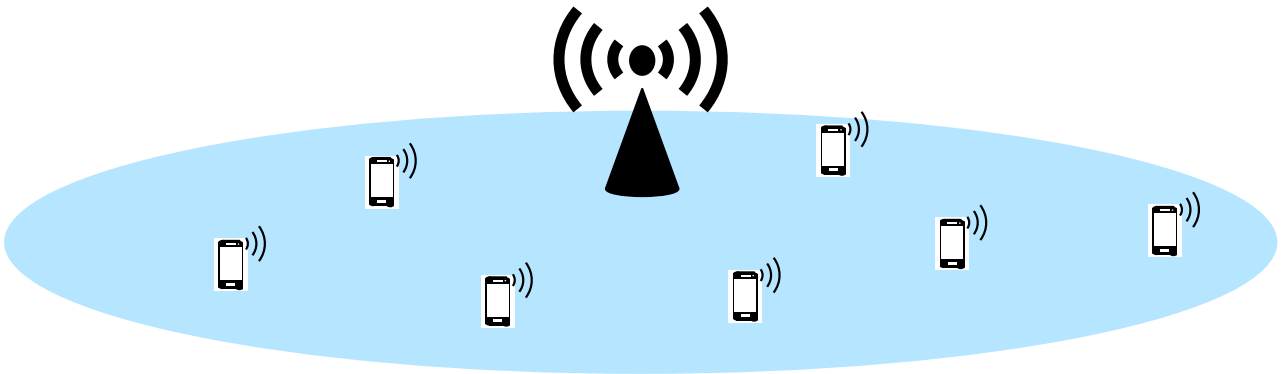
Source: Cisco VNI Mobile, 2014

Number of devices



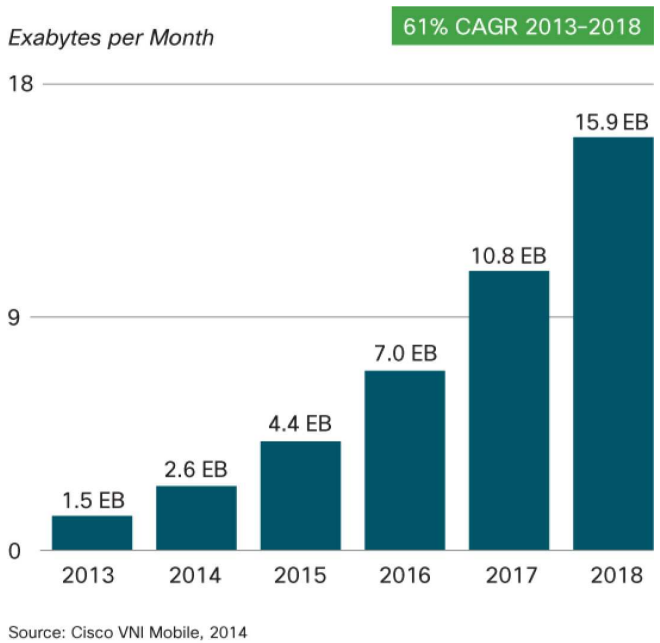
Percentages refer to device or connections share.

Source: Cisco VNI Mobile, 2014



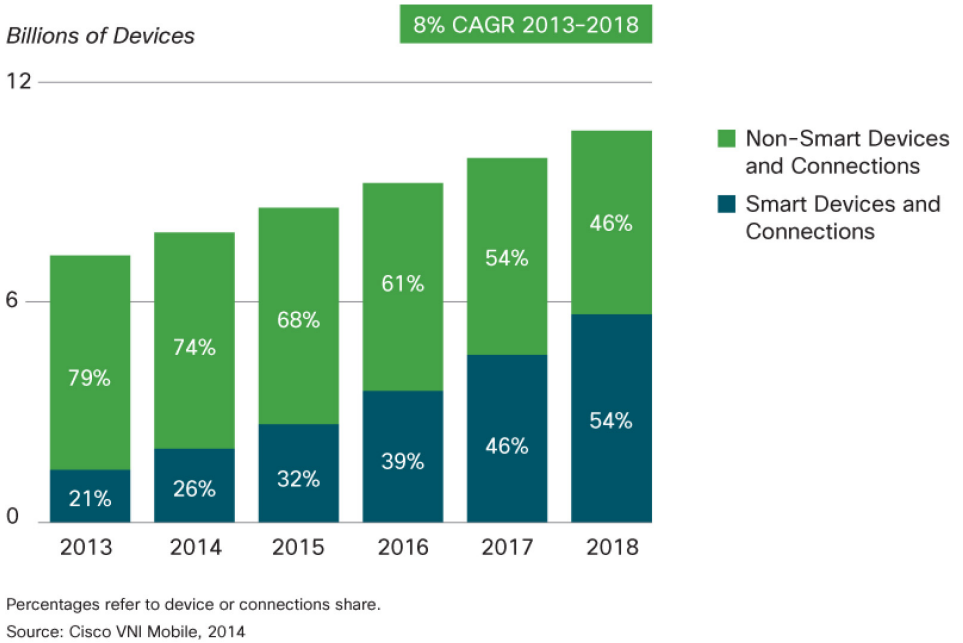
Where is wireless going?

Mobile data per month



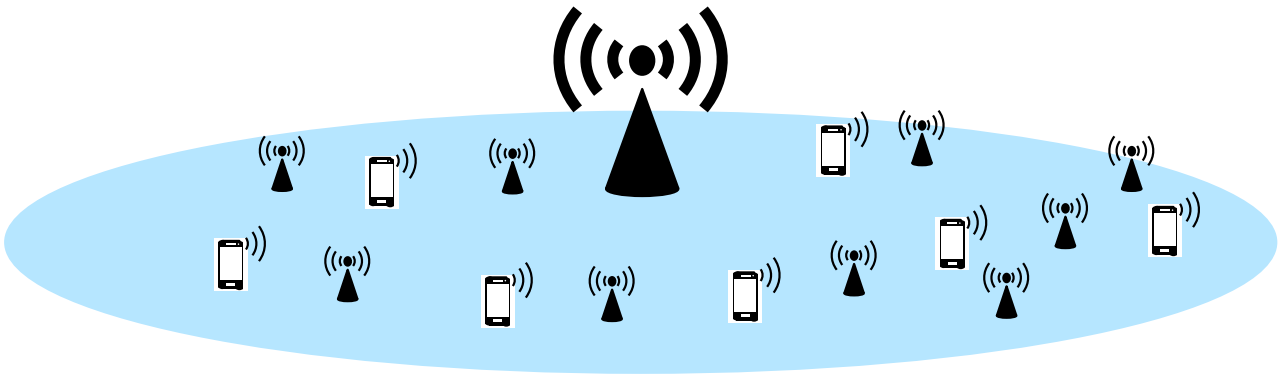
Source: Cisco VNI Mobile, 2014

Number of devices



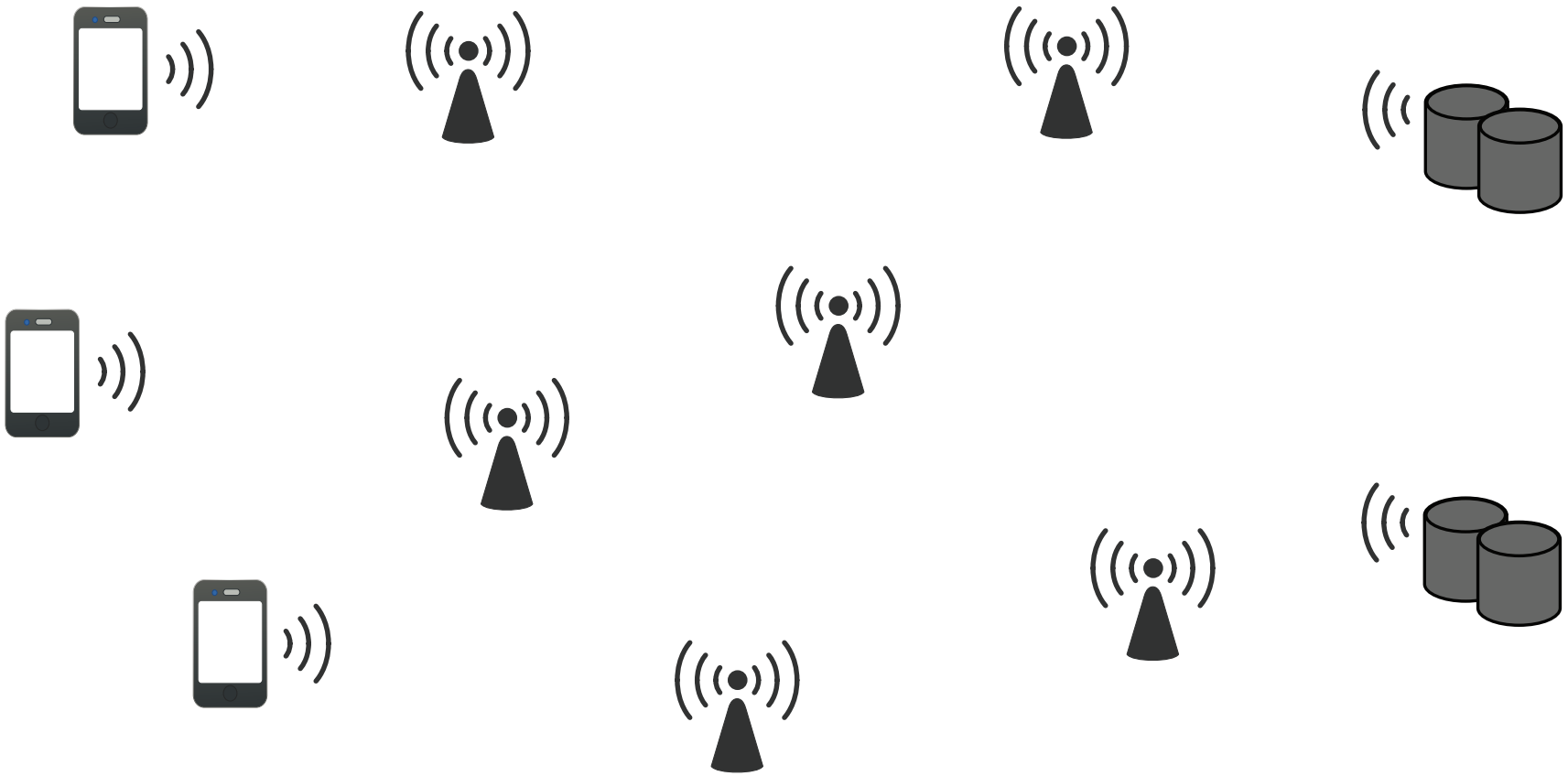
Percentages refer to device or connections share.

Source: Cisco VNI Mobile, 2014

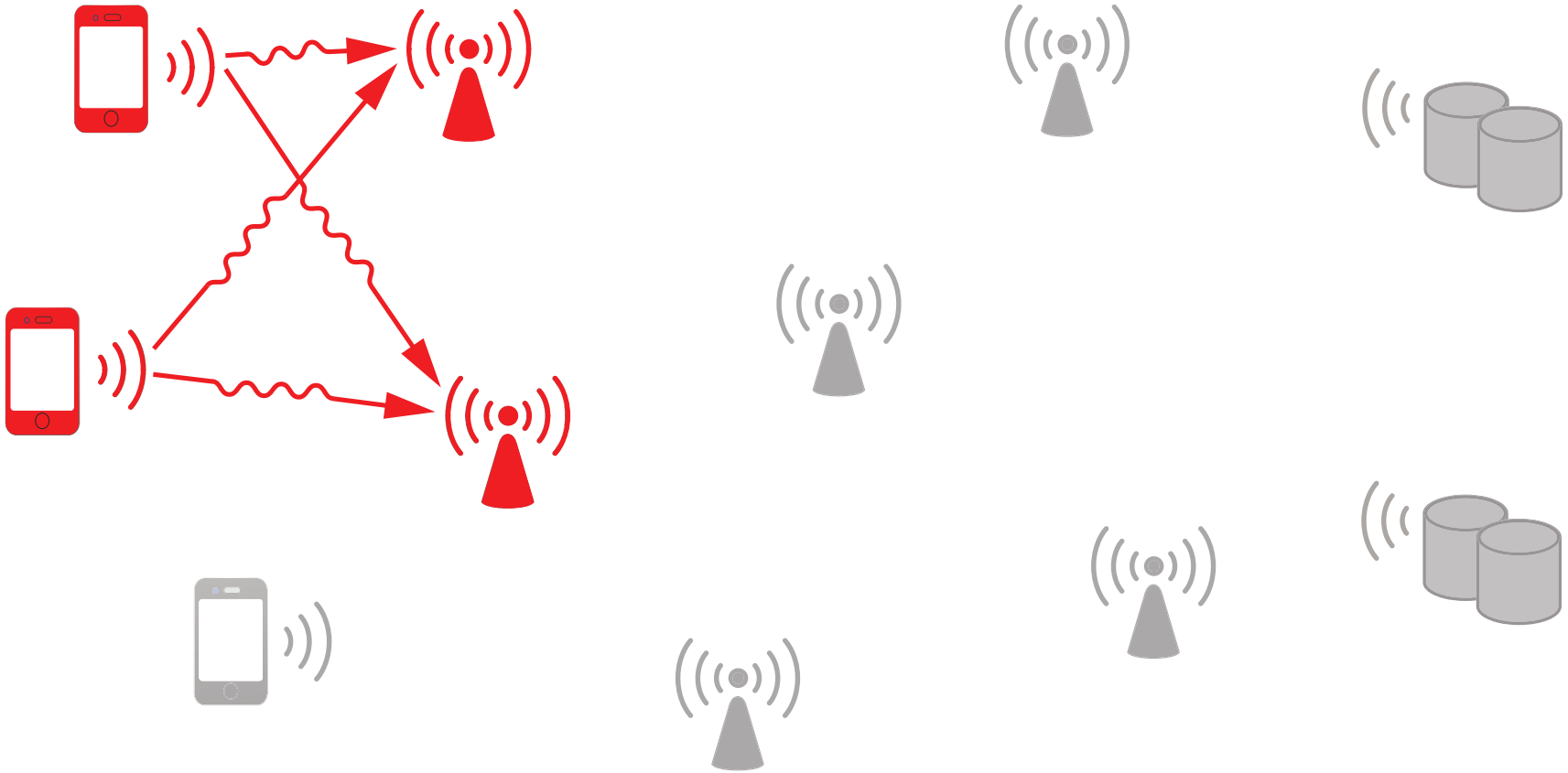



The Internet of THINGS
CONNECTED OBJECTS
CONNECTED PLANT
3km range
4
CONNECT THE WORLD

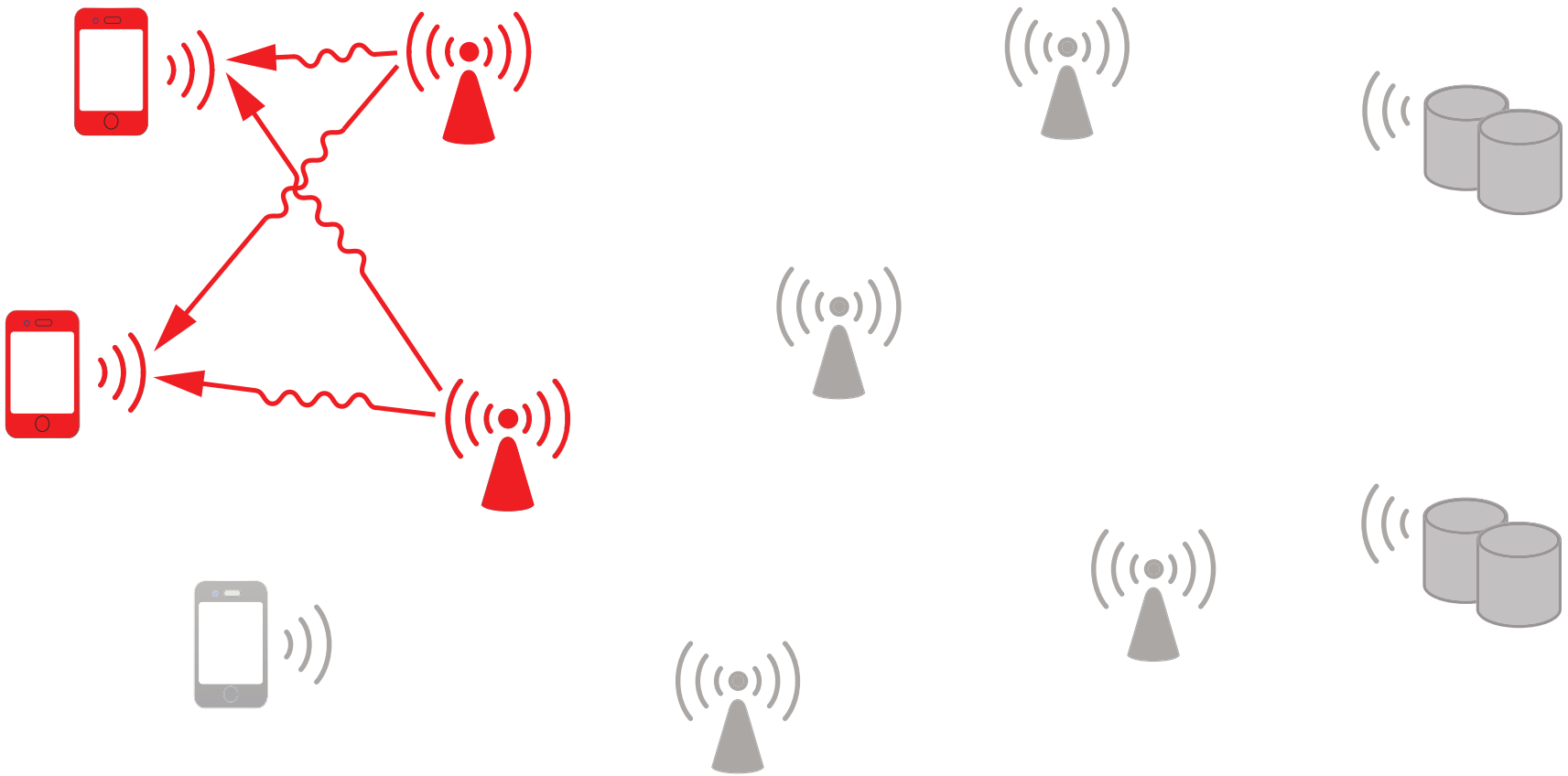
Interference, interference, interference



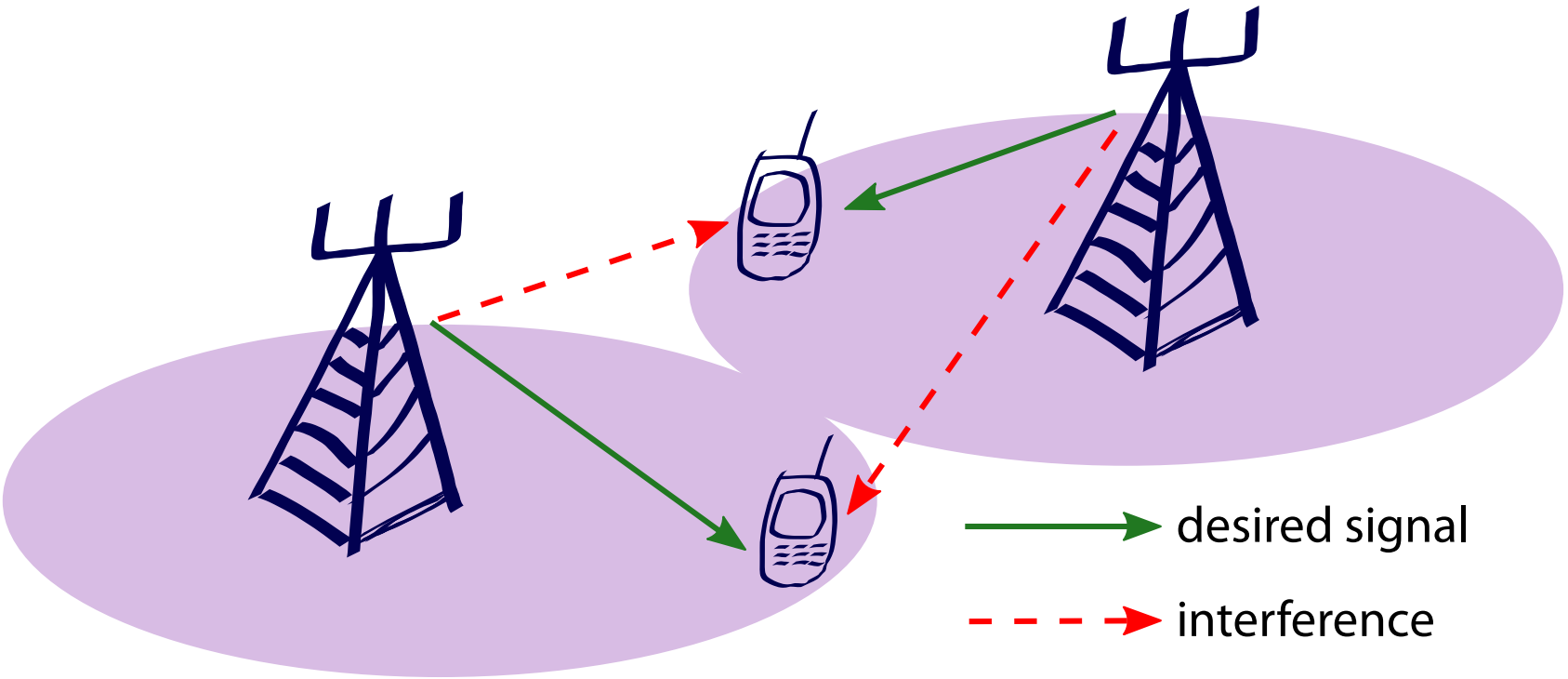
Interference, interference, interference



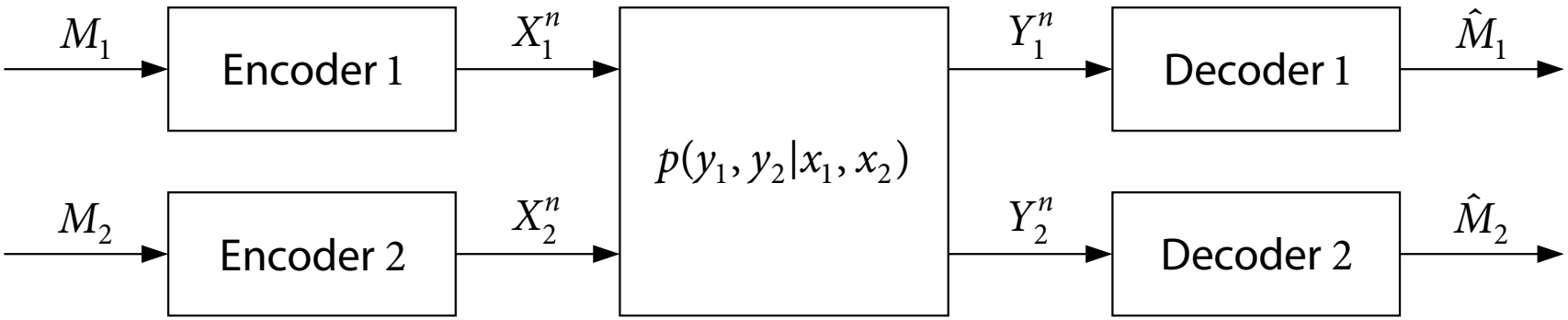
Interference, interference, interference



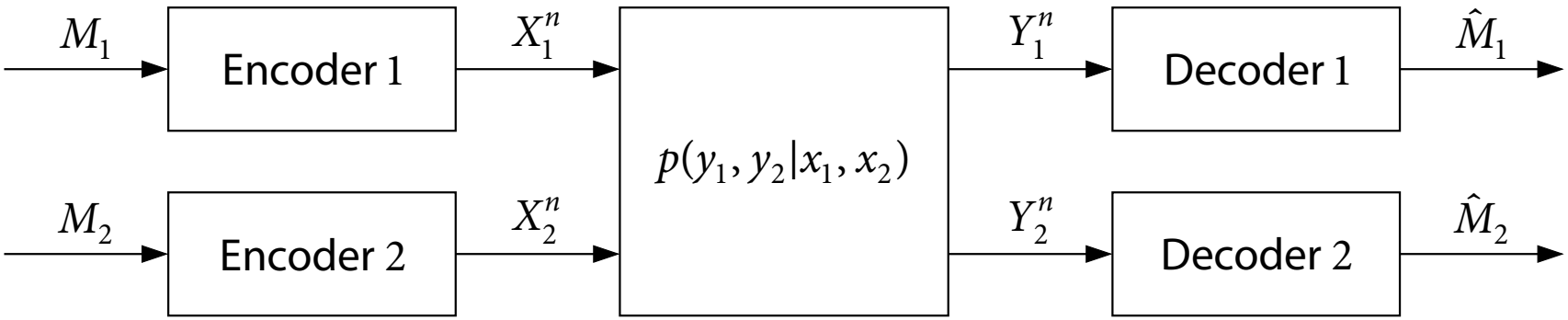
Interference in cellular networks



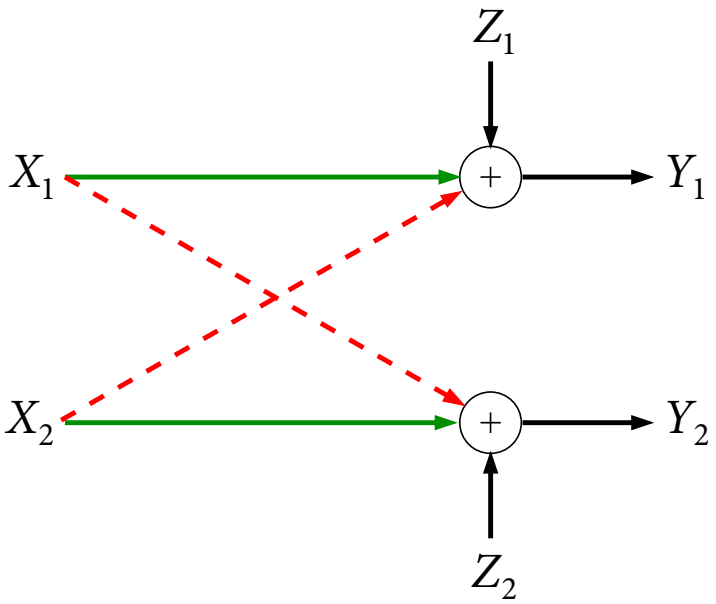
Interference channel



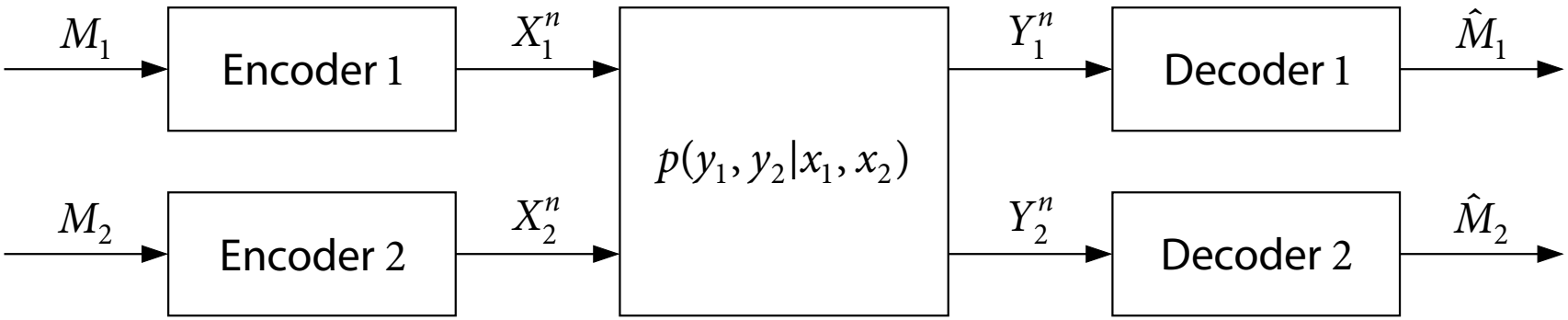
Interference channel



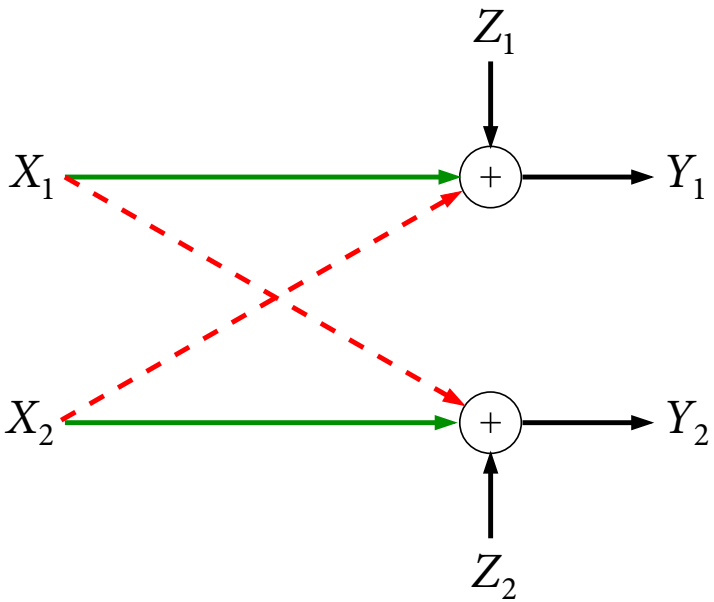
- Gaussian interference channel



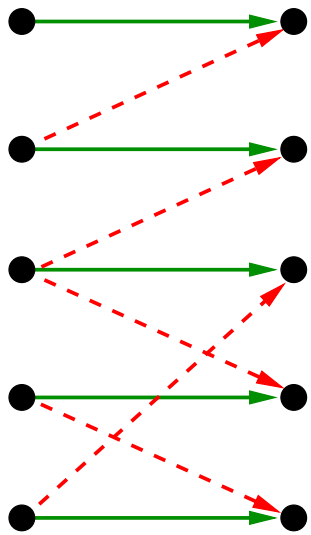
Interference channel



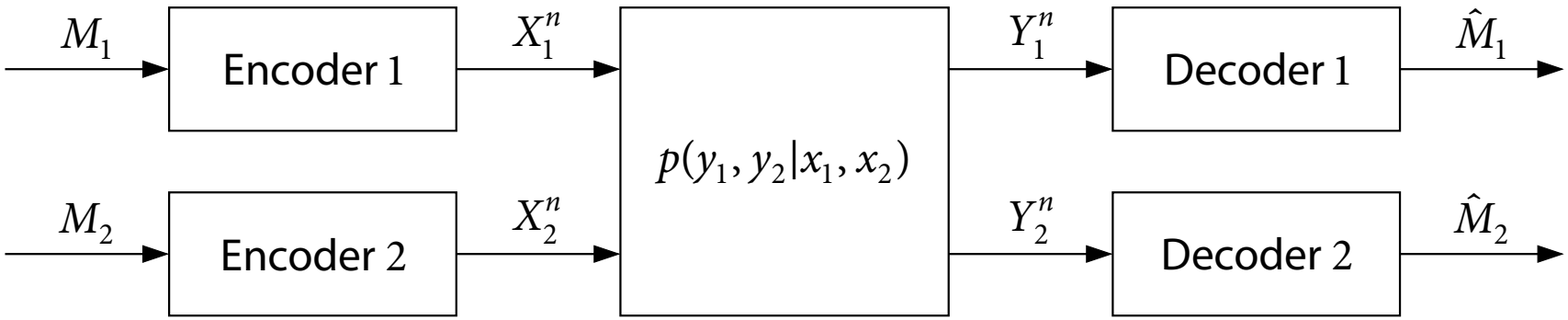
- Gaussian interference channel



- Network with one dominant interferer



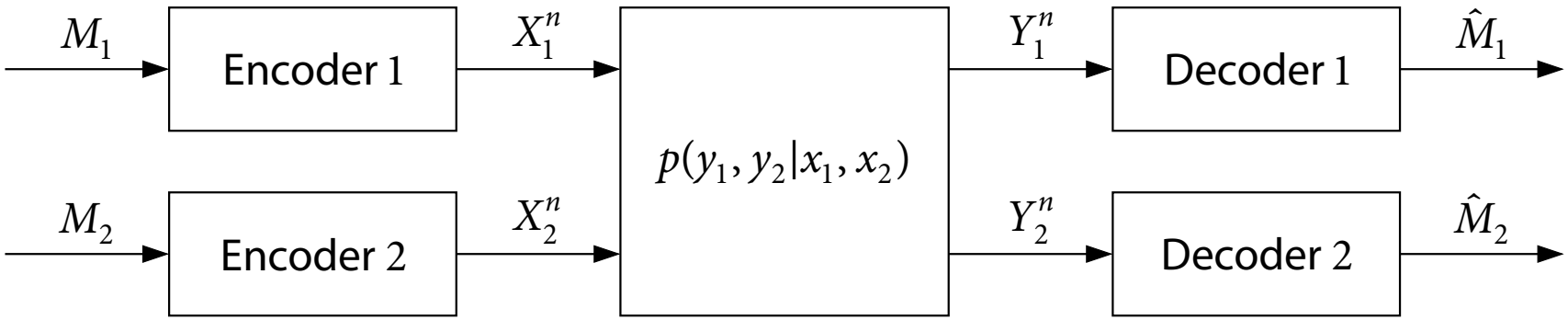
Interference channel



- Capacity region: Optimal tradeoff between the rates

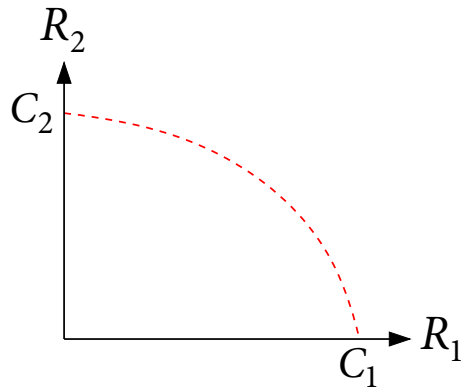
$$R_1 = \frac{1}{n} \log |\text{supp}(M_1)|$$
$$R_2 = \frac{1}{n} \log |\text{supp}(M_2)|$$

Interference channel

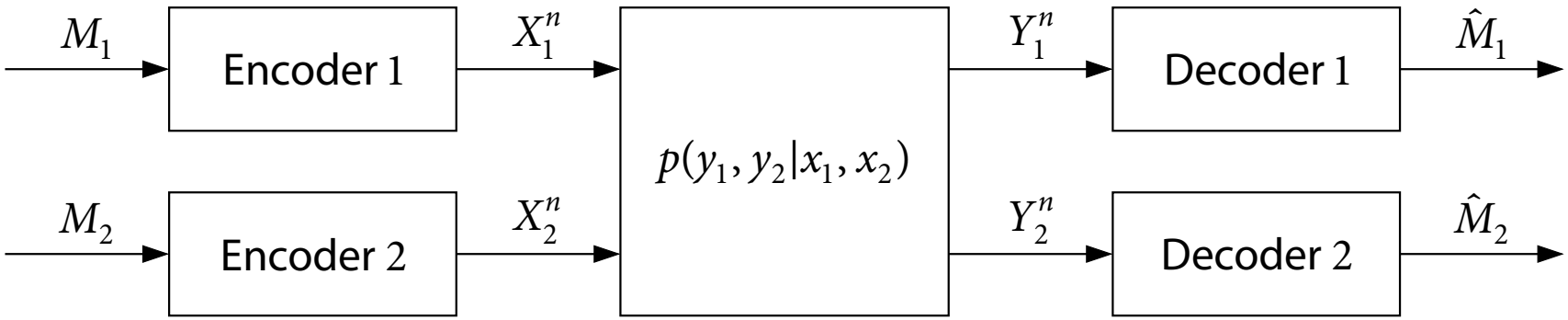


- Capacity region: Optimal tradeoff between the rates

$$R_1 = \frac{1}{n} \log |\text{supp}(M_1)|$$
$$R_2 = \frac{1}{n} \log |\text{supp}(M_2)|$$

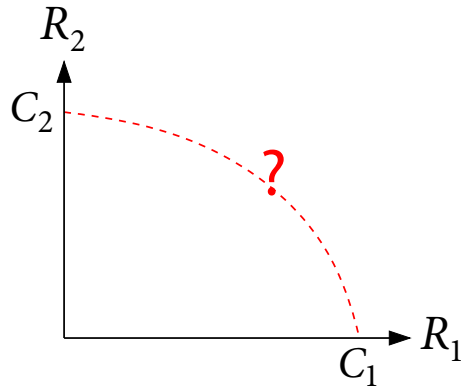


Interference channel

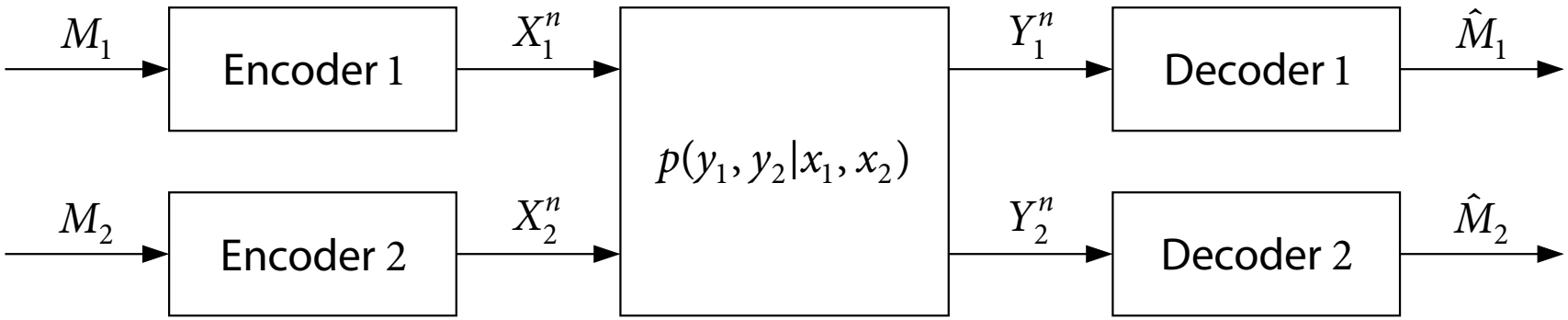


- Capacity region: Optimal tradeoff between the rates

$$R_1 = \frac{1}{n} \log |\text{supp}(M_1)|$$
$$R_2 = \frac{1}{n} \log |\text{supp}(M_2)|$$

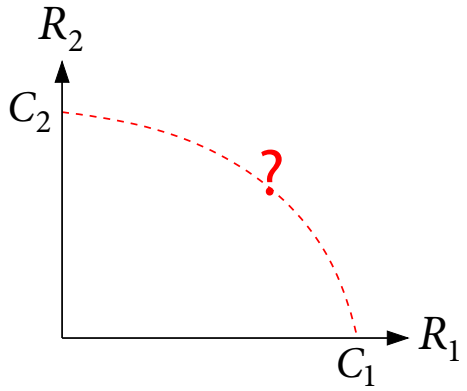


Interference channel



- Capacity region: Optimal tradeoff between the rates

$$R_1 = \frac{1}{n} \log |\mathrm{supp}(M_1)|$$
$$R_2 = \frac{1}{n} \log |\mathrm{supp}(M_2)|$$



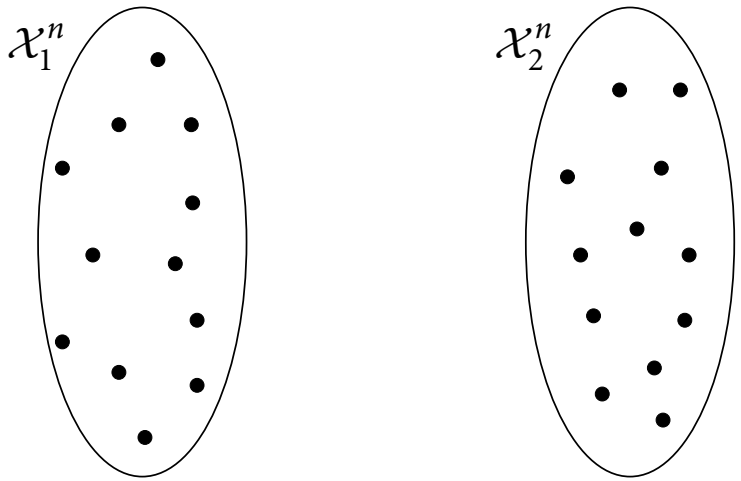
- Best inner bound: Han–Kobayashi (1981) and Nair–Xia–Yazdanpanah (2015)

Performance benchmark

- Highest rates achievable by point-to-point (P2P) random code ensembles

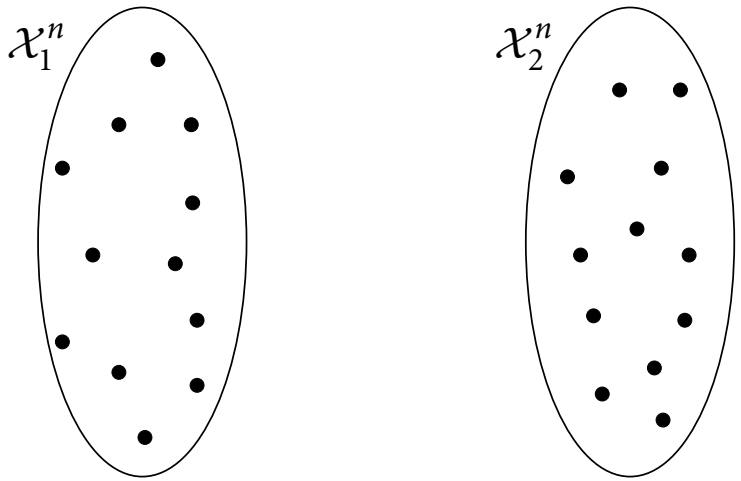
Performance benchmark

- Highest rates achievable by point-to-point (P2P) random code ensembles
- Random code ensemble
  - For each $m_1 \in [1 : 2^{nR_1}]$, generate $X_1^n(m_1) \sim \prod_{i=1}^n p_{X_1}(x_{1i})$
  - For each $m_2 \in [1 : 2^{nR_2}]$, generate $X_2^n(m_2) \sim \prod_{i=1}^n p_{X_2}(x_{2i})$



Performance benchmark

- Highest rates achievable by point-to-point (P2P) random code ensembles
- Random code ensemble
  - For each $m_1 \in [1 : 2^{nR_1}]$, generate $X_1^n(m_1) \sim \prod_{i=1}^n p_{X_1}(x_{1i})$
  - For each $m_2 \in [1 : 2^{nR_2}]$, generate $X_2^n(m_2) \sim \prod_{i=1}^n p_{X_2}(x_{2i})$



- What is the optimal (MLD) tradeoff between achievable R_1 and R_2?

Why do we care?

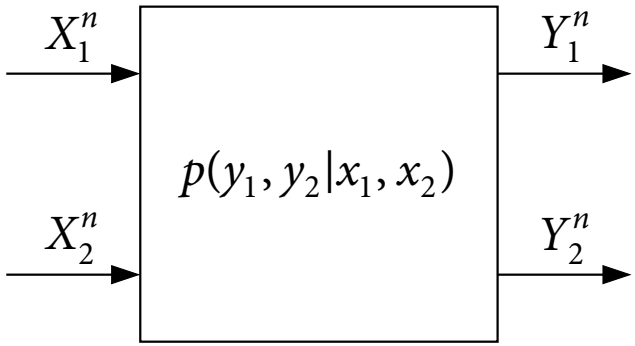
- Optimal when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)

Why do we care?

- Optimal when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
- P2P random codes ≈ COTS (commercial off-the-shelf) codes

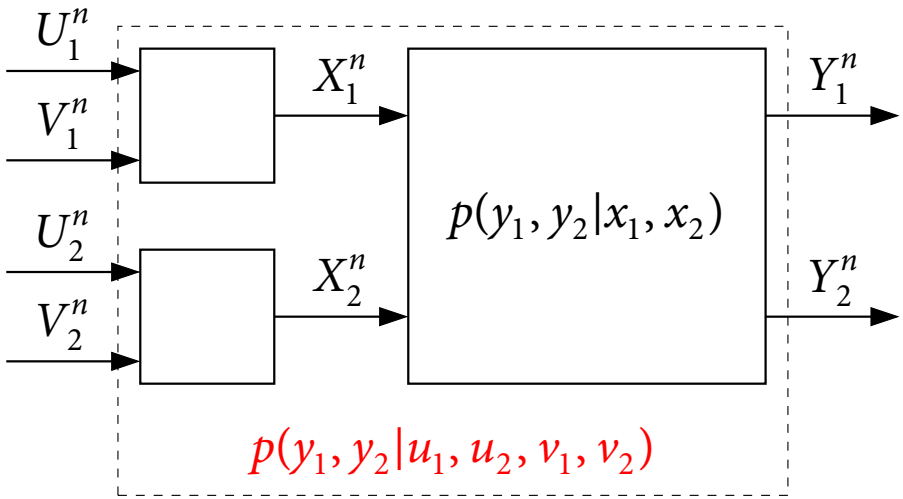
Why do we care?

- Optimal when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
- P2P random codes ≈ COTS (commercial off-the-shelf) codes
- Han–Kobayashi coding scheme



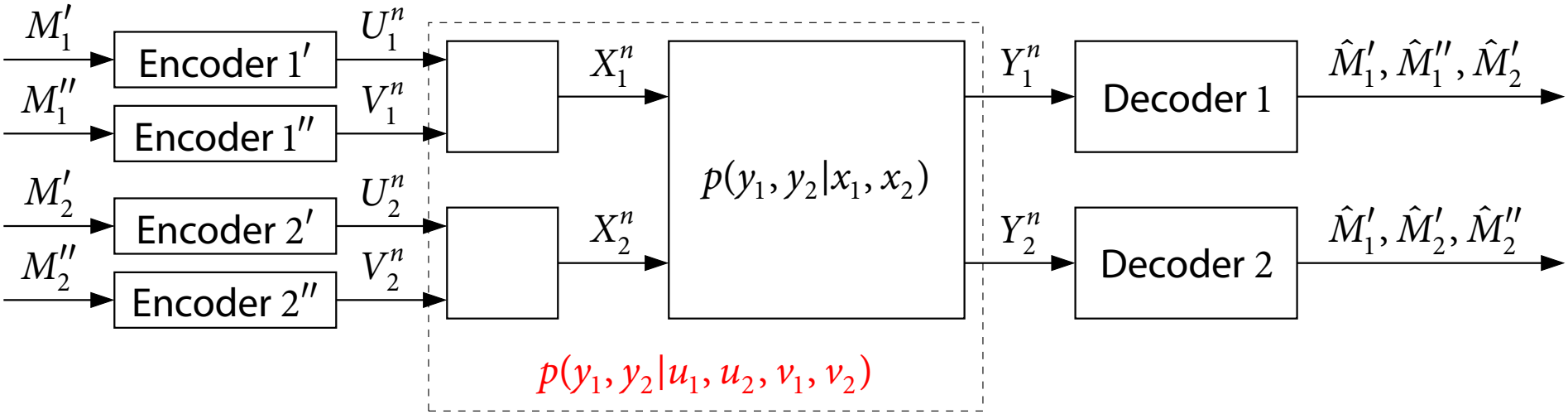
Why do we care?

- Optimal when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
- P2P random codes ≈ COTS (commercial off-the-shelf) codes
- Han–Kobayashi coding scheme

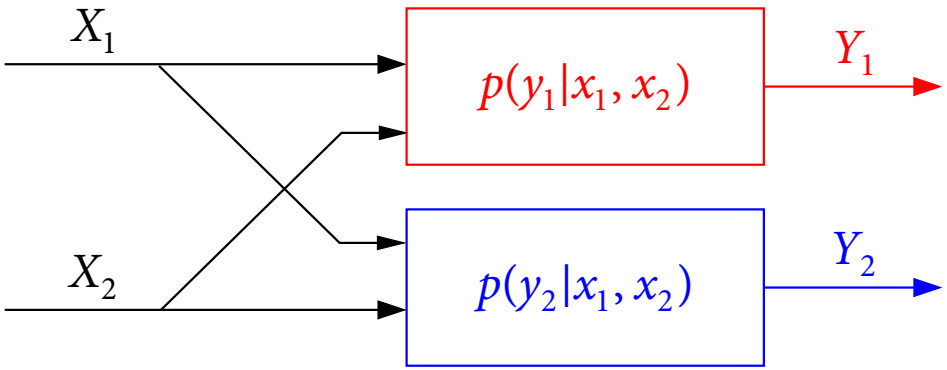


Why do we care?

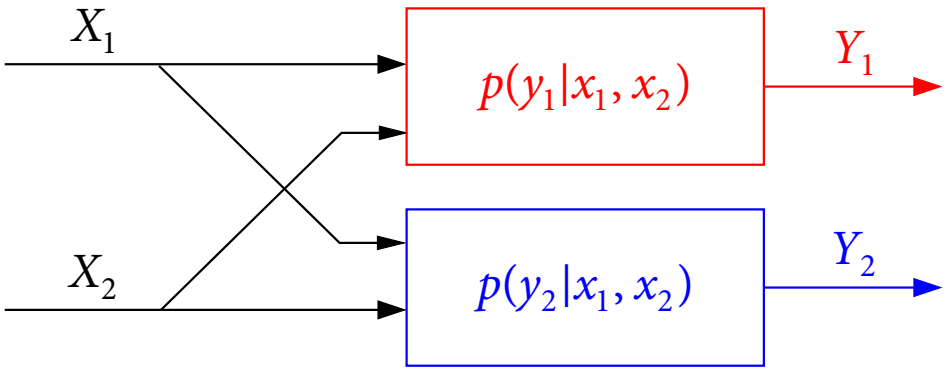
- Optimal when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
- P2P random codes ≈ COTS (commercial off-the-shelf) codes
- Han–Kobayashi coding scheme



Performance benchmark (Bandemer–El-Gamal–K 2012)



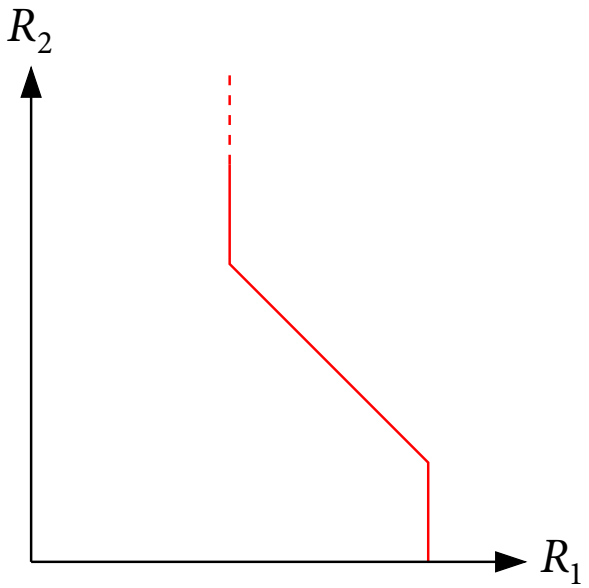
Performance benchmark (Bandemer–El-Gamal–K 2012)



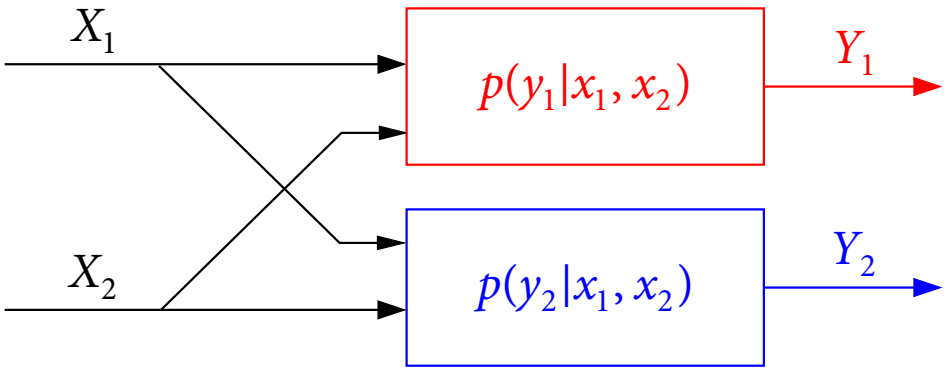
$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$



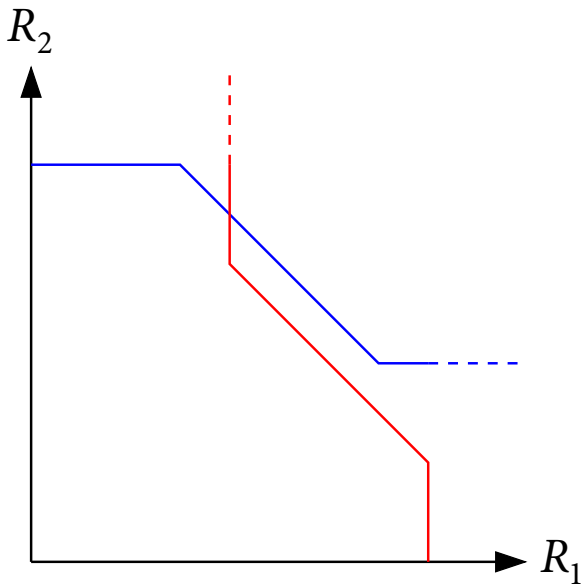
Performance benchmark (Bandemer–El-Gamal–K 2012)



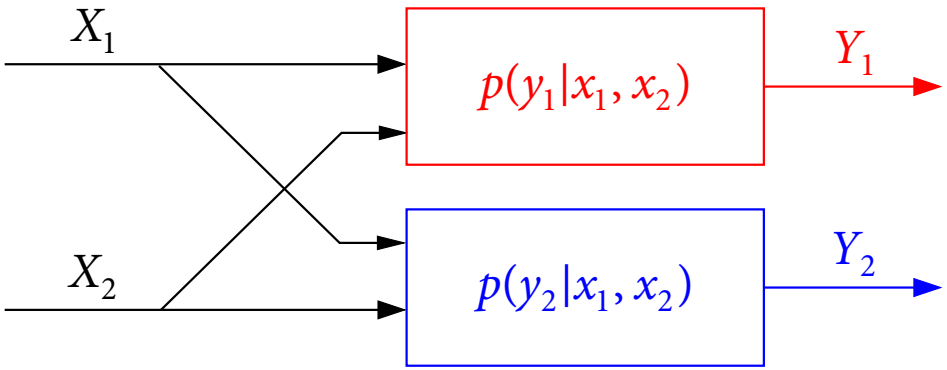
$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$



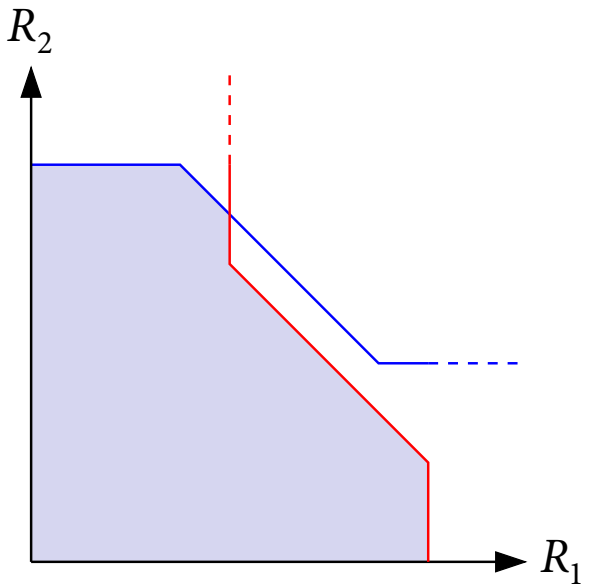
Performance benchmark (Bandemer–El-Gamal–K 2012)



$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$

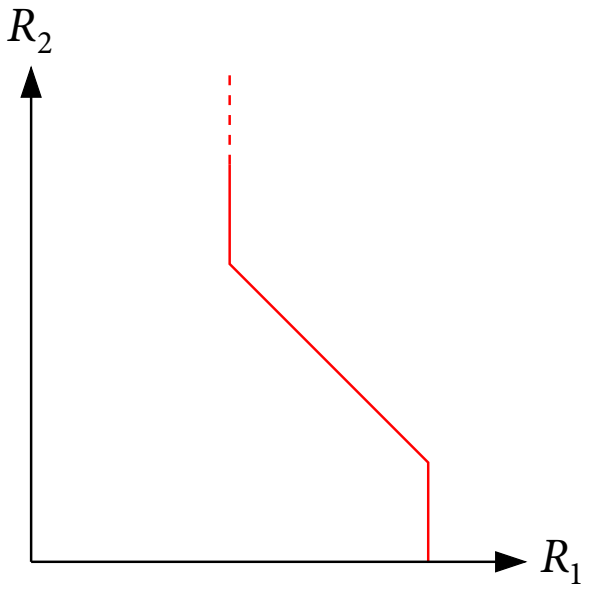


Maximum likelihood decoding ≈ simultaneous decoding

$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$

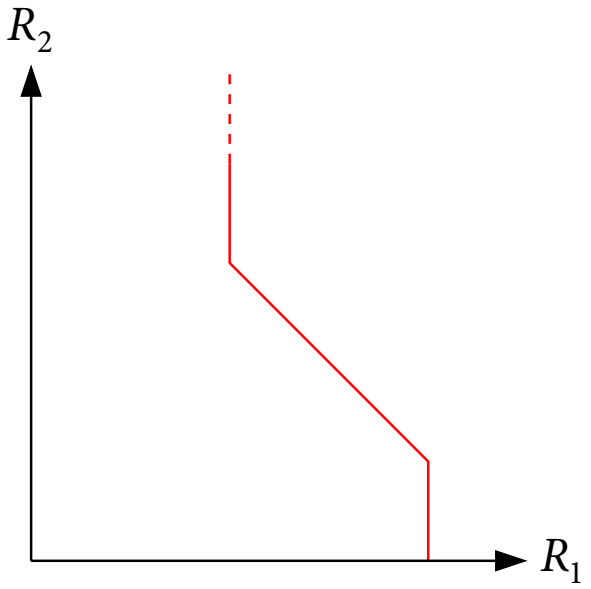


Maximum likelihood decoding ≈ simultaneous decoding

$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$



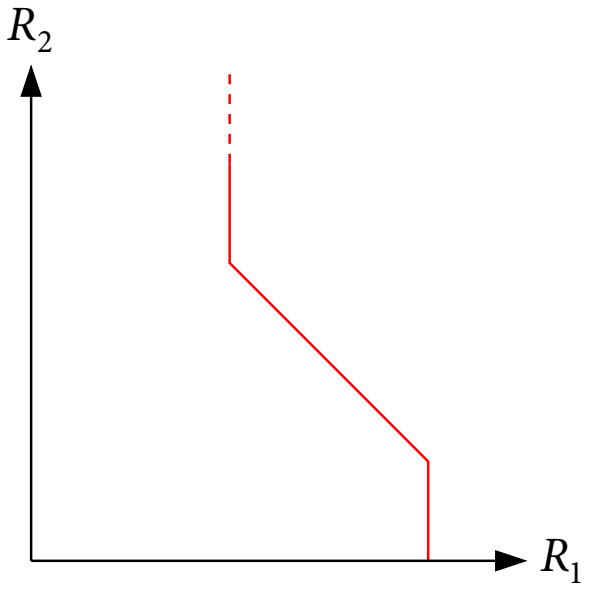
- Maximum likelihood decoding: $\arg\max_{\hat{m}_1} \sum_{m_2} p(y_1^n | \hat{m}_1, m_2)$

Maximum likelihood decoding ≈ simultaneous decoding

$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$



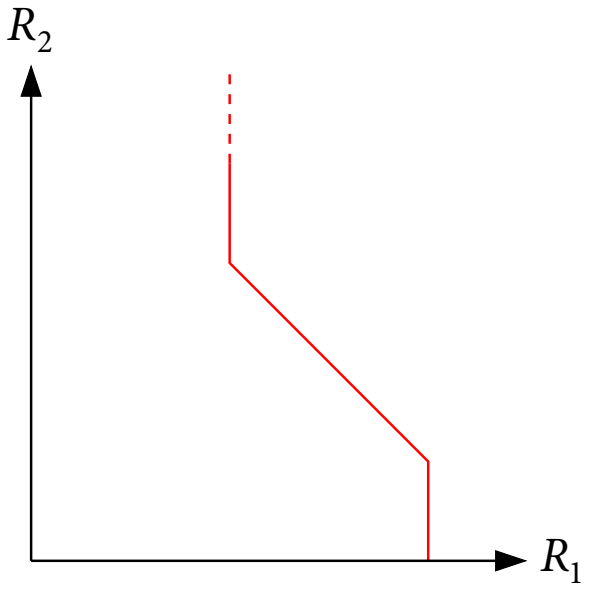
- Maximum likelihood decoding: $\arg\max_{\hat{m}_1} \sum_{m_2} p(y_1^n | \hat{m}_1, m_2)$
- Simultaneous ML decoding: $\arg\max_{\hat{m}_1} \max_{m_2} p(y_1^n | \hat{m}_1, m_2)$

Maximum likelihood decoding ≈ simultaneous decoding

$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

or

$$R_1 < I(X_1; Y_1)$$



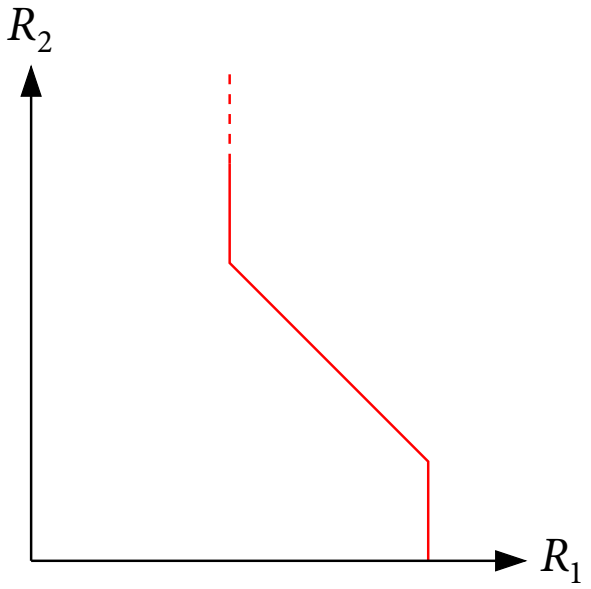
- Maximum likelihood decoding: $\arg\max_{\hat{m}_1} \sum_{m_2} p(y_1^n | \hat{m}_1, m_2)$
- Simultaneous ML decoding: $\arg\max_{\hat{m}_1} \max_{m_2} p(y_1^n | \hat{m}_1, m_2)$
- Simultaneous nonunique decoding (SND): $(x_1^n(\hat{m}_1), x_2^n(m_2), y_1^n) \in \mathcal{T}_\epsilon^{(n)}$ for some m_2

Maximum likelihood decoding ≈ simultaneous decoding

$$R_1 < I(X_1; Y_1|X_2),$$
$$R_1 + R_2 < I(X_1, X_2; Y_1)$$

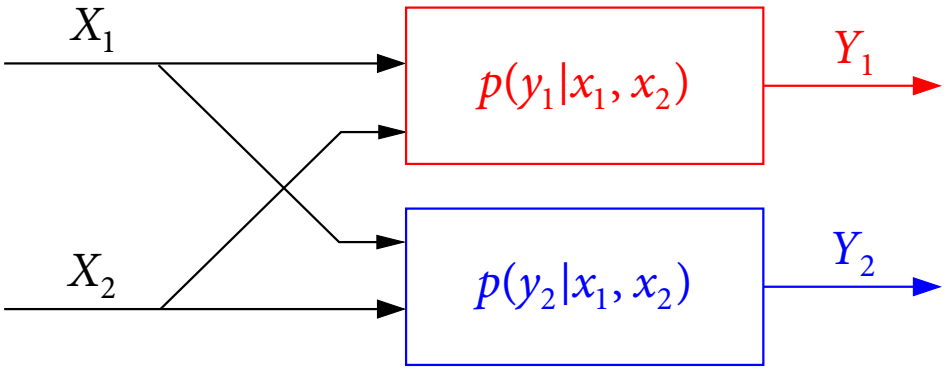
or

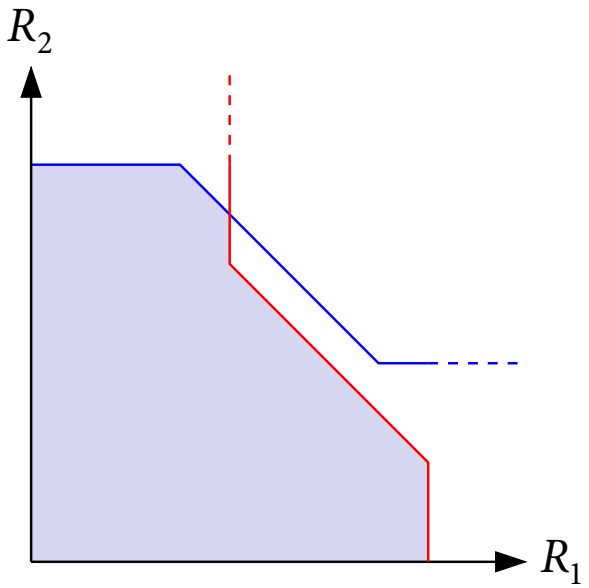
$$R_1 < I(X_1; Y_1)$$



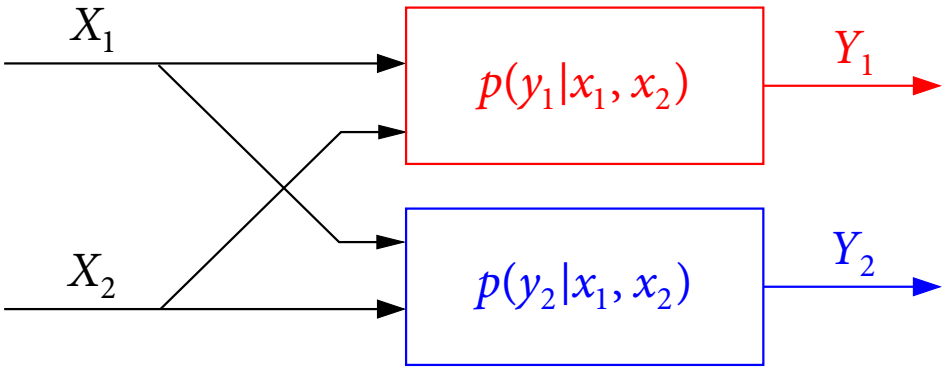
- Maximum likelihood decoding: $\arg\max_{\hat{m}_1} \sum_{m_2} p(y_1^n|\hat{m}_1, m_2)$
- Simultaneous ML decoding: $\arg\max_{\hat{m}_1} \max_{m_2} p(y_1^n|\hat{m}_1, m_2)$
- Simultaneous nonunique decoding (SND): $(x_1^n(\hat{m}_1), x_2^n(m_2), y_1^n) \in \mathcal{T}_\epsilon^{(n)}$ for some m_2
- Multiuser detection: High complexity!

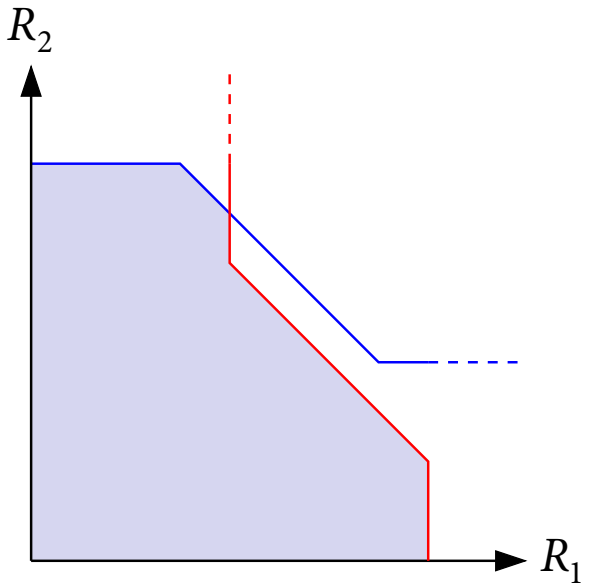
Low-complexity (implementable) alternatives





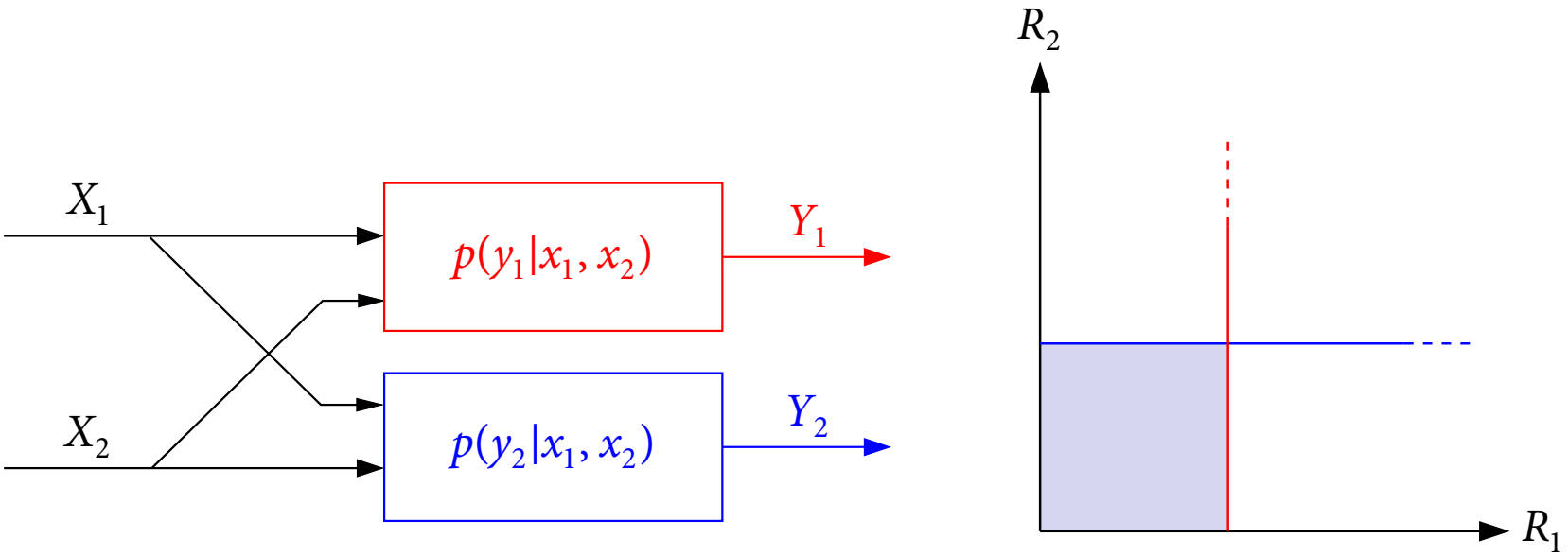
Low-complexity (implementable) alternatives





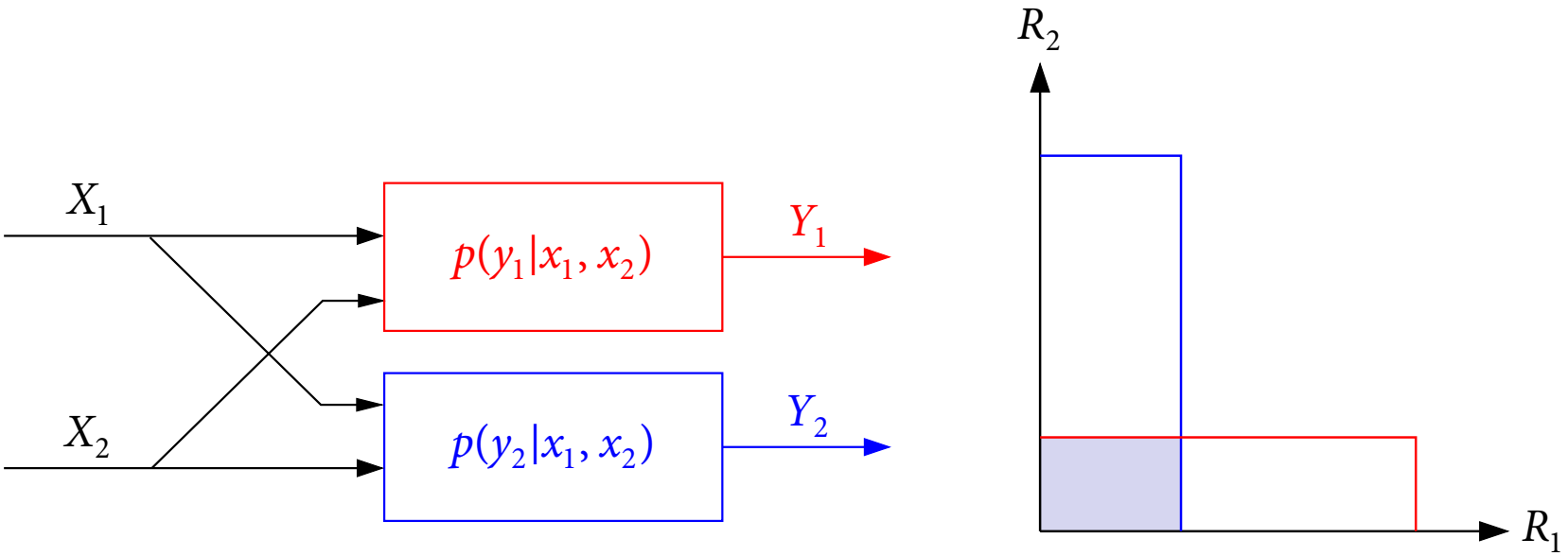
- P2P decoding

Low-complexity (implementable) alternatives



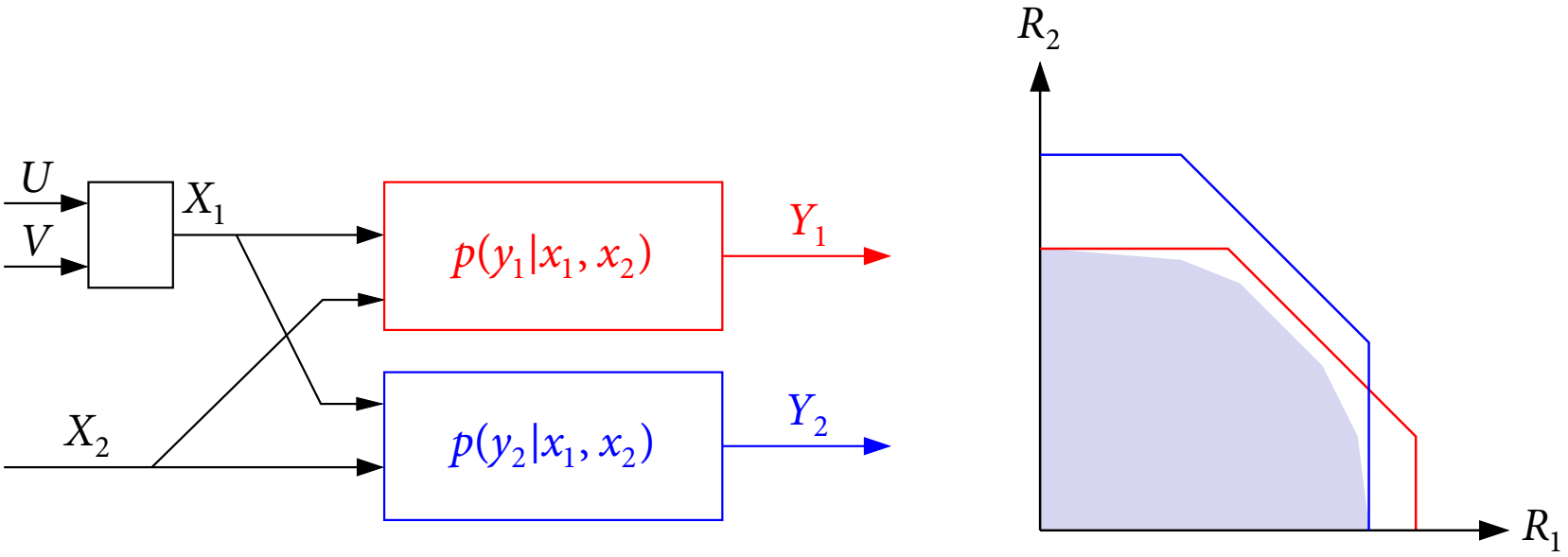
- P2P decoding
  - Treating interference as (Gaussian) noise: $R_1 < I(X_1; Y_1)$

Low-complexity (implementable) alternatives



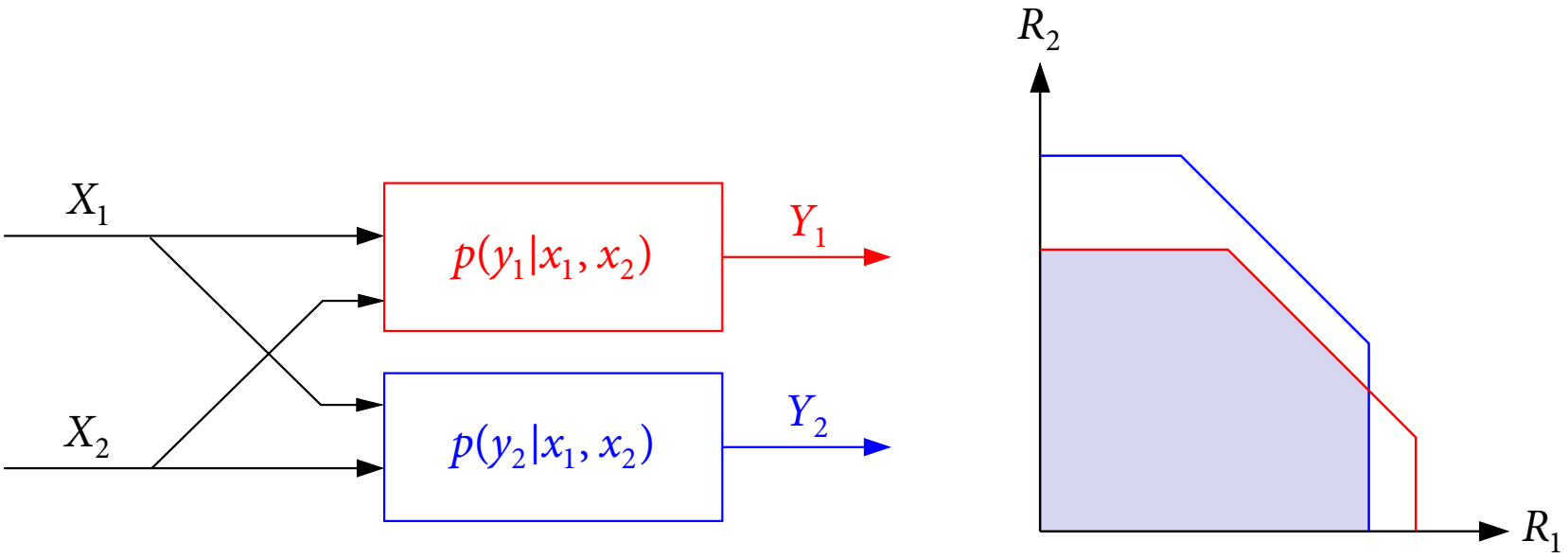
- P2P decoding
  - Treating interference as (Gaussian) noise: $R_1 < I(X_1; Y_1)$
  - Successive cancellation decoding: $R_2 < I(X_2; Y_1),\ R_1 < I(X_1; Y_1|X_2)$

Low-complexity (implementable) alternatives



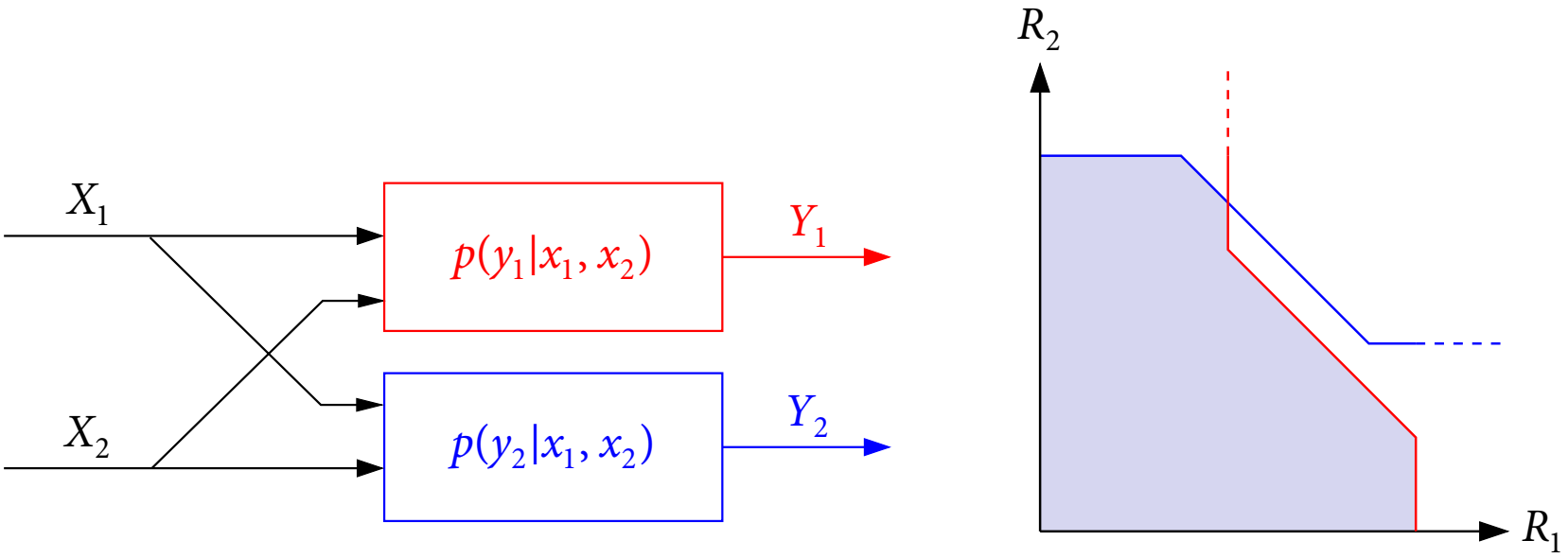
- P2P decoding
  - Treating interference as (Gaussian) noise: $R_1 < I(X_1; Y_1)$
  - Successive cancellation decoding: $R_2 < I(X_2; Y_1),\ R_1 < I(X_1; Y_1|X_2)$
- + rate splitting (Zhao et al. 2011, Wang et al. 2014)

Low-complexity (implementable) alternatives



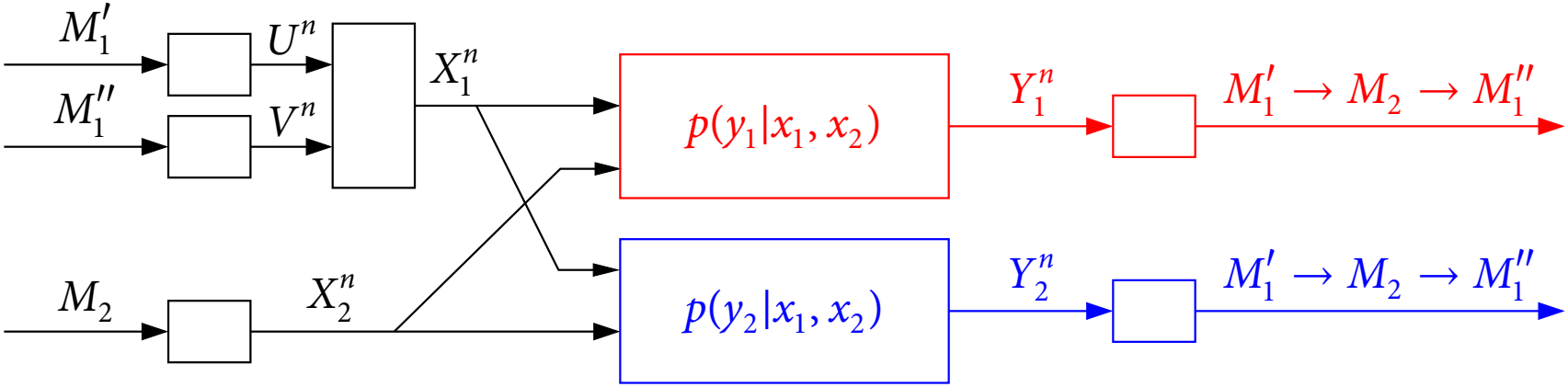
- P2P decoding
  - Treating interference as (Gaussian) noise: $R_1 < I(X_1; Y_1)$
  - Successive cancellation decoding: $R_2 < I(X_2; Y_1)$, $R_1 < I(X_1; Y_1|X_2)$
- + rate splitting (Zhao et al. 2011, Wang et al. 2014)
- Novel codes
  - Spatially coupled codes (Yedla, Nguyen, Pfister, and Narayanan 2011)
  - Polar codes (Wang and Şaşoğlu 2014)

Low-complexity (implementable) alternatives

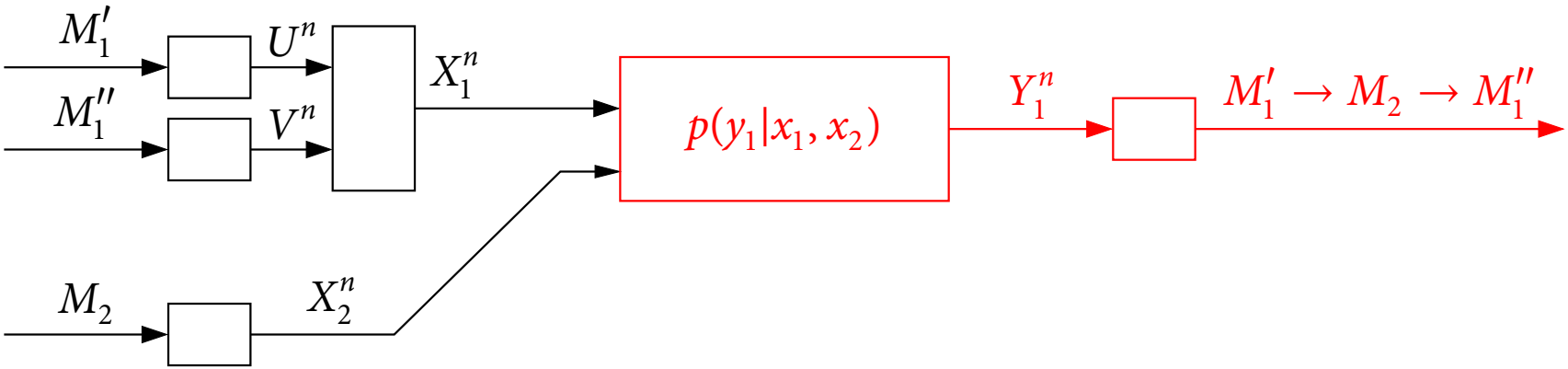


- P2P decoding
  - Treating interference as (Gaussian) noise: $R_1 < I(X_1; Y_1)$
  - Successive cancellation decoding: $R_2 < I(X_2; Y_1)$, $R_1 < I(X_1; Y_1|X_2)$
- + rate splitting (Zhao et al. 2011, Wang et al. 2014)
- Novel codes
  - Spatially coupled codes (Yedla, Nguyen, Pfister, and Narayanan 2011)
  - Polar codes (Wang and Şaşoğlu 2014)

A lesson from rate splitting (Grant et al. 2001)

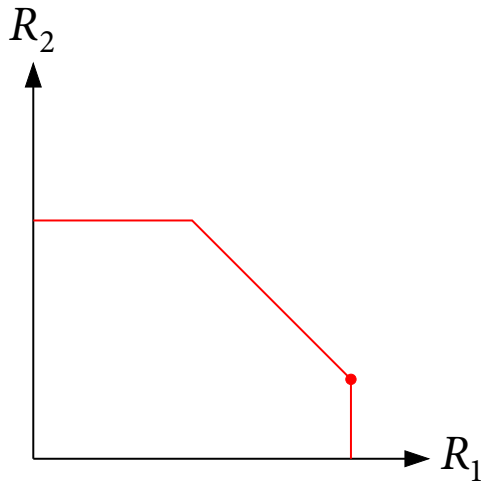


A lesson from rate splitting (Grant et al. 2001)

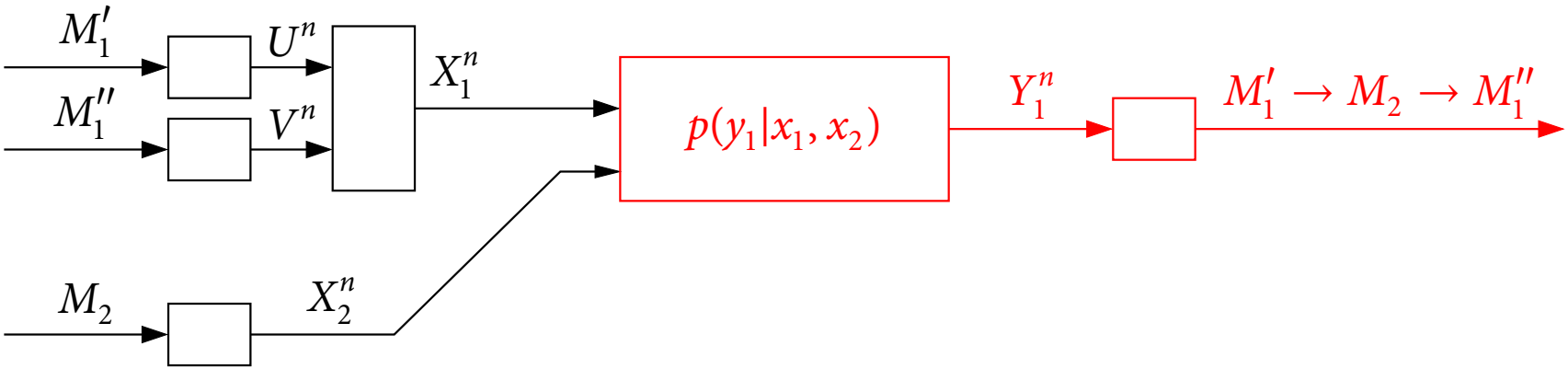


- Decoding at receiver 1:

$$R_1' < I(U; Y_1)$$
$$R_2 < I(X_2; Y_1|U)$$
$$R_1'' < I(V; Y_1|U, X_2)$$

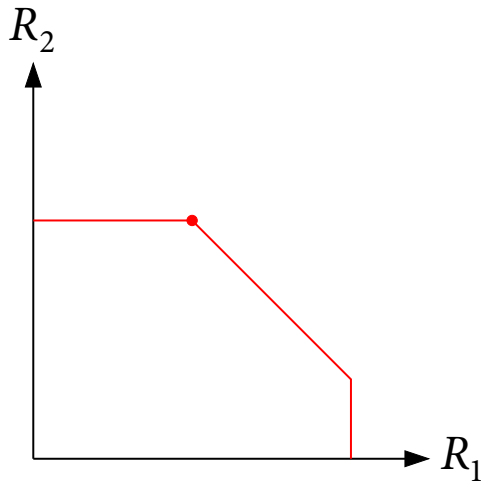


A lesson from rate splitting (Grant et al. 2001)

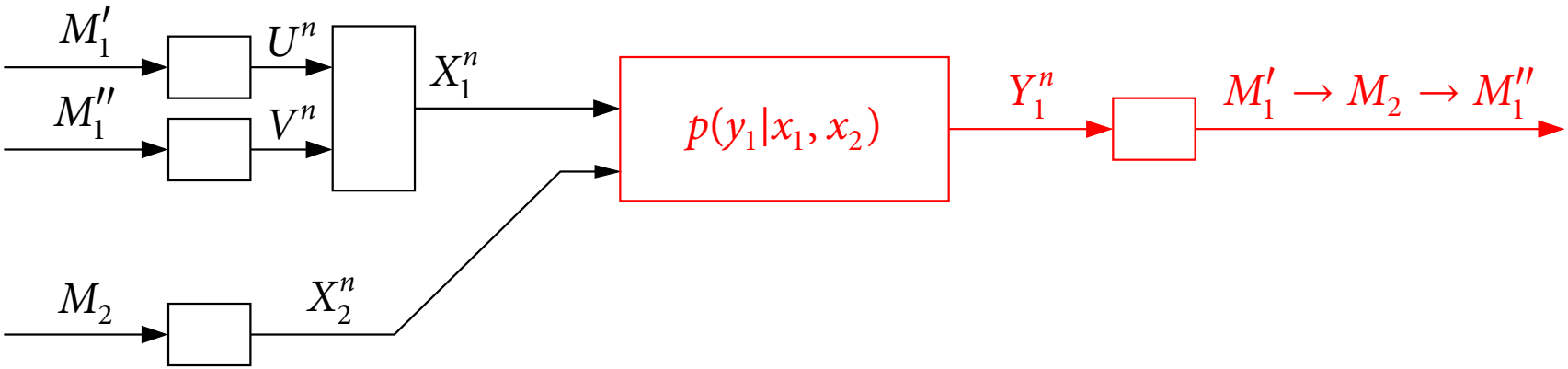


- Decoding at receiver 1:

$$R_1' < I(U; Y_1)$$
$$R_2 < I(X_2; Y_1|U)$$
$$R_1'' < I(V; Y_1|U, X_2)$$

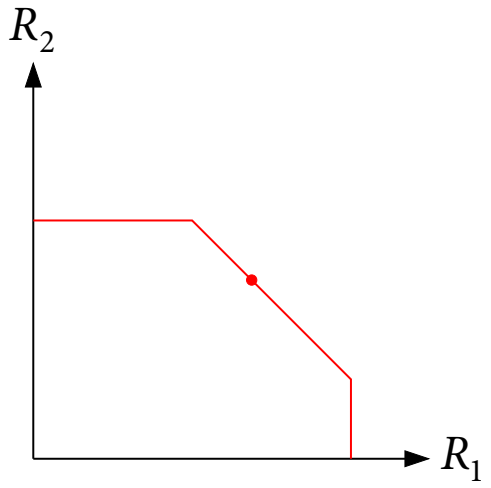


A lesson from rate splitting (Grant et al. 2001)

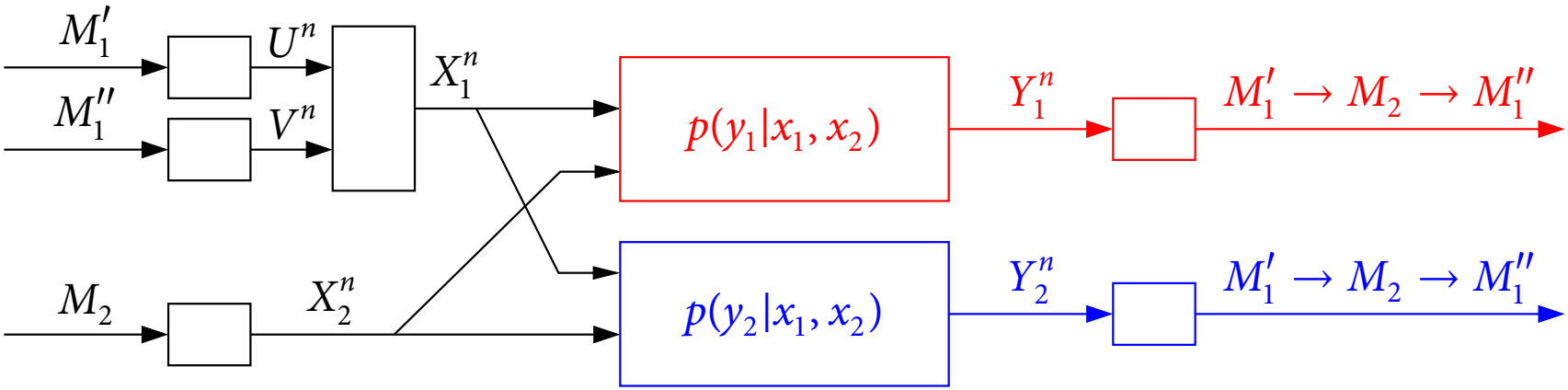


- Decoding at receiver 1:

$$R_1' < I(U; Y_1)$$
$$R_2 < I(X_2; Y_1|U)$$
$$R_1'' < I(V; Y_1|U, X_2)$$



A lesson from rate splitting (Grant et al. 2001)

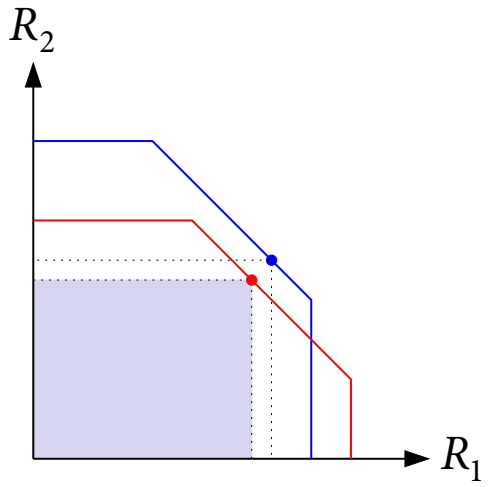


- Decoding at receiver 1:

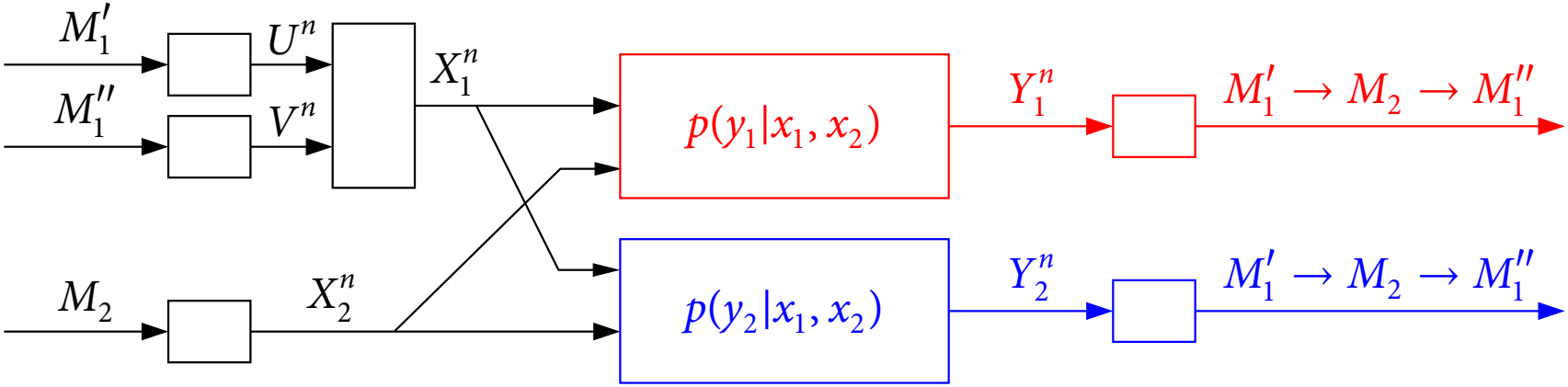
$$R_1' < I(U; Y_1)$$
$$R_2 < I(X_2; Y_1|U)$$
$$R_1'' < I(V; Y_1|U, X_2)$$

- Decoding at receiver 2:

$$R_1' < I(U; Y_2)$$
$$R_2 < I(X_2; Y_2|U)$$
$$R_1'' < I(V; Y_2|U, X_2)$$



A lesson from rate splitting (Grant et al. 2001)

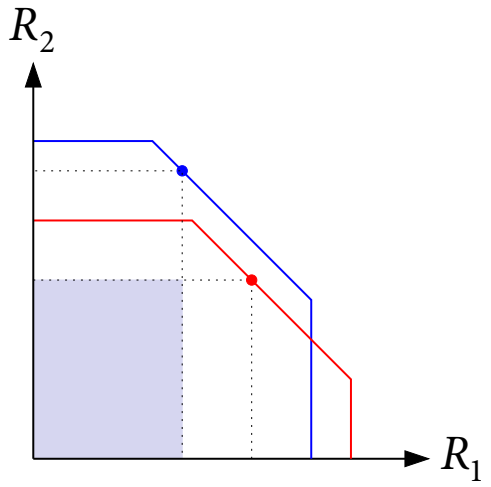


- Decoding at receiver 1:

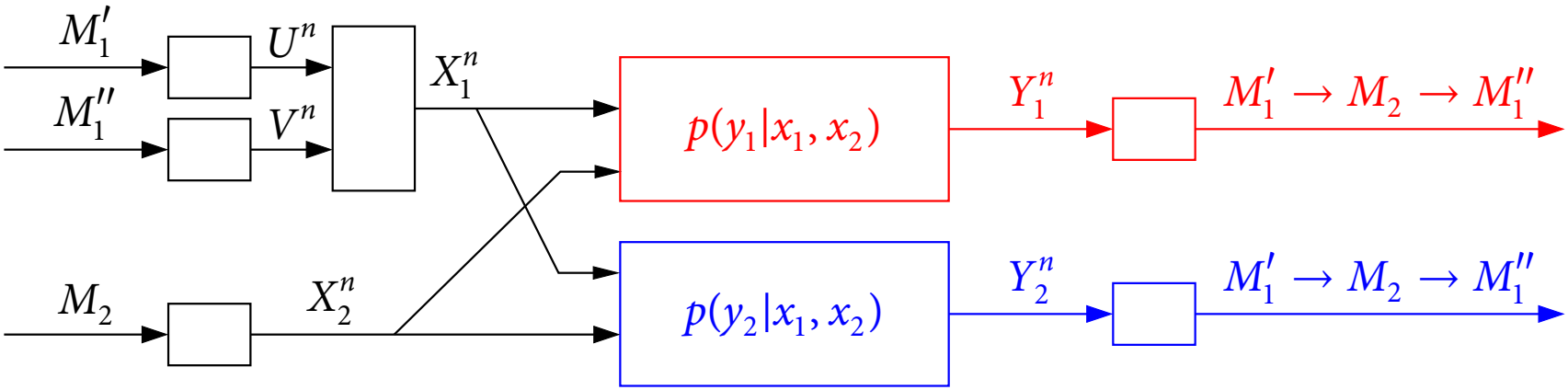
$$R_1' < I(U; Y_1)$$
$$R_2 < I(X_2; Y_1|U)$$
$$R_1'' < I(V; Y_1|U, X_2)$$

- Decoding at receiver 2:

$$R_1' < I(U; Y_2)$$
$$R_2 < I(X_2; Y_2|U)$$
$$R_1'' < I(V; Y_2|U, X_2)$$

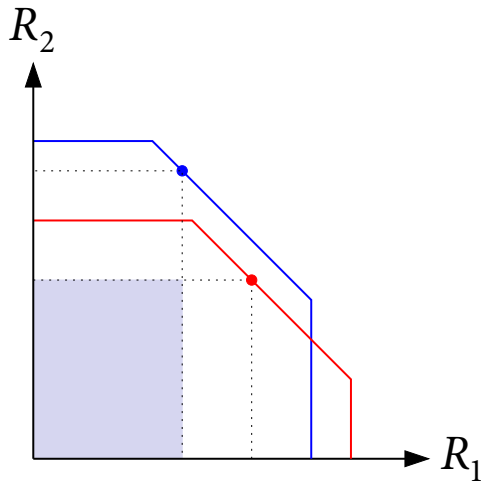


A lesson from rate splitting (Grant et al. 2001)

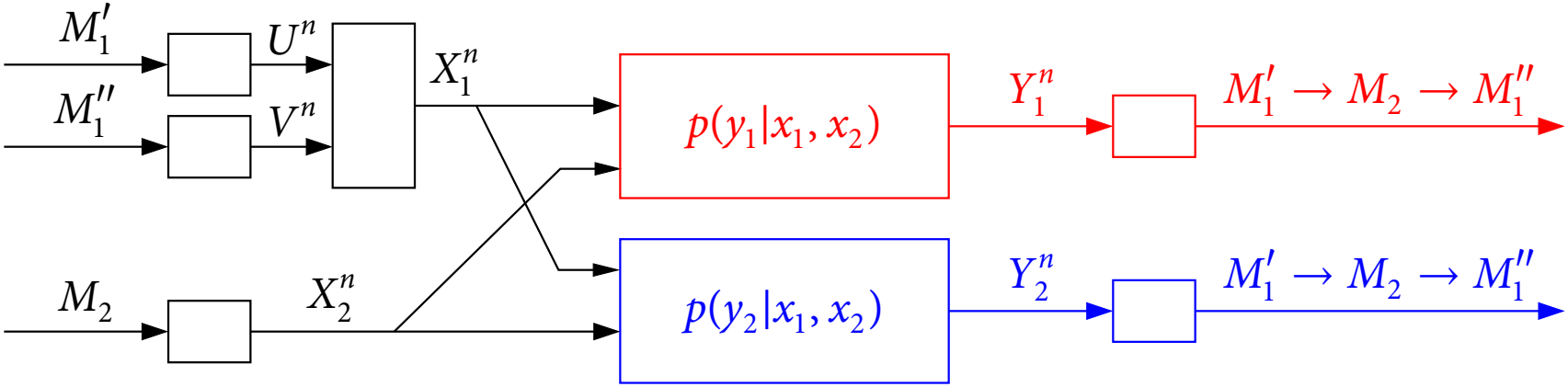


- Decoding at receiver 1: $R_1' < I(U; Y_1),\ R_2 < I(X_2; Y_1|U),\ R_1'' < I(V; Y_1|U, X_2)$
- Decoding at receiver 2: $R_1' < I(U; Y_2),\ R_2 < I(X_2; Y_2|U),\ R_1'' < I(V; Y_2|U, X_2)$
- Combined rate:

$$R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2)$$

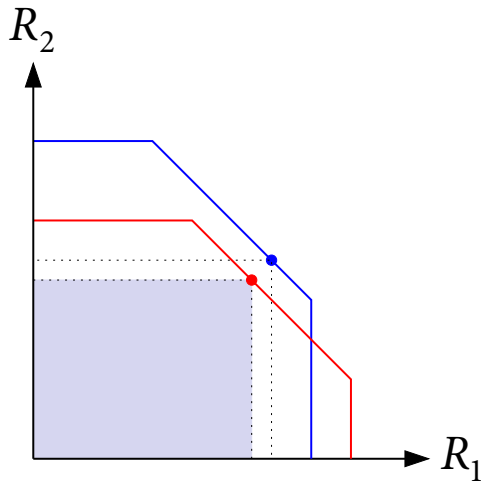


A lesson from rate splitting (Grant et al. 2001)

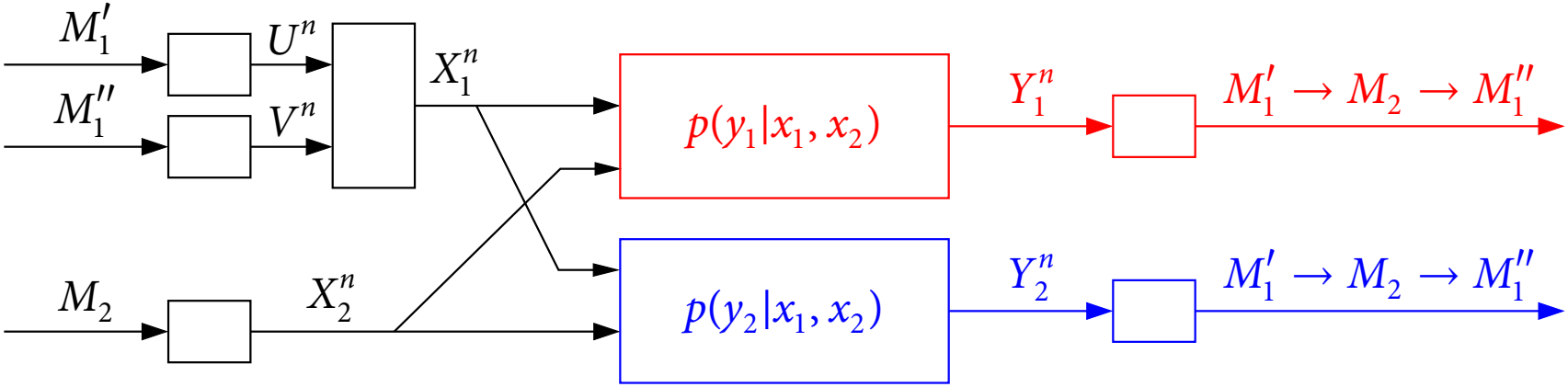


- Decoding at receiver 1: $R_1' < I(U; Y_1),\ R_2 < I(X_2; Y_1|U),\ R_1'' < I(V; Y_1|U, X_2)$
- Decoding at receiver 2: $R_1' < I(U; Y_2),\ R_2 < I(X_2; Y_2|U),\ R_1'' < I(V; Y_2|U, X_2)$
- Combined rate:

$$R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2)$$
$$< \min_j [I(U; Y_j) + I(V; Y_j|U, X_2)]$$



A lesson from rate splitting (Grant et al. 2001)

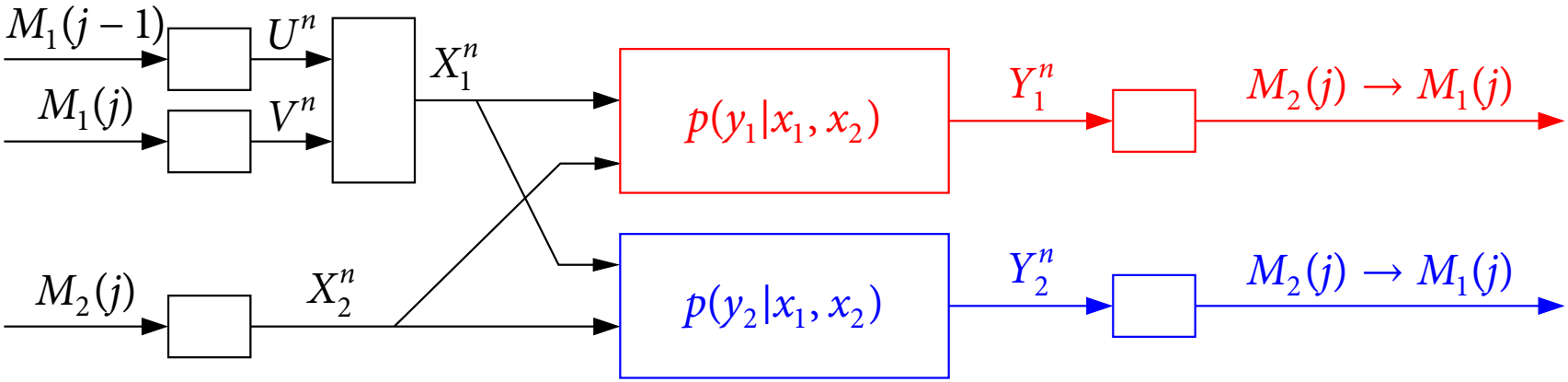


- Decoding at receiver 1: $R_1' < I(U; Y_1),\ R_2 < I(X_2; Y_1|U),\ R_1'' < I(V; Y_1|U, X_2)$
- Decoding at receiver 2: $R_1' < I(U; Y_2),\ R_2 < I(X_2; Y_2|U),\ R_1'' < I(V; Y_2|U, X_2)$
- Combined rate:

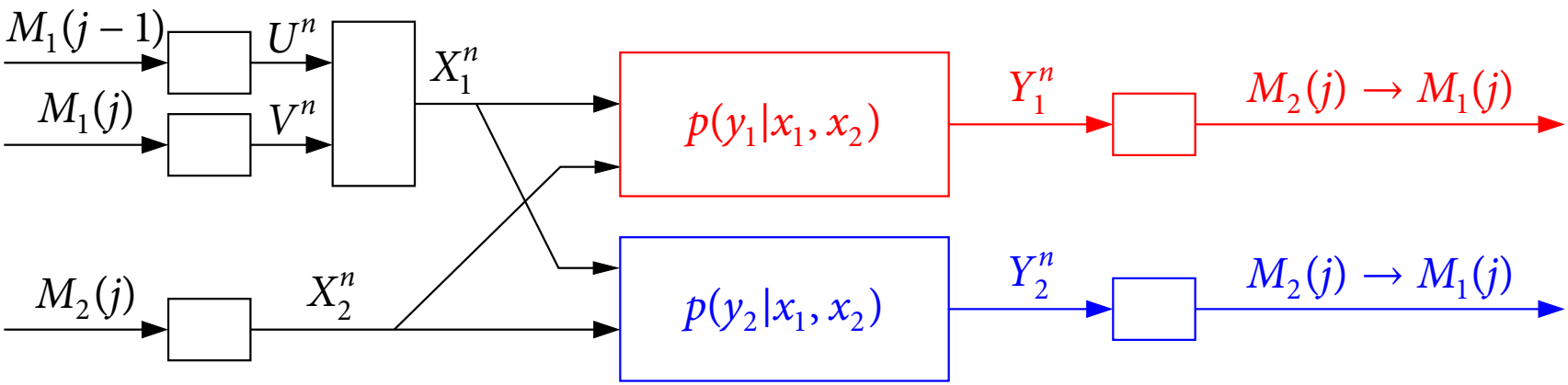
$$R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2)$$
$$< \min_j \left[I(U; Y_j) + I(V; Y_j|U, X_2) \right]$$

- Key to achieving the SD performance: Switch the order of sum and min!

Sliding-window superposition coding (Wang et al. 2014)

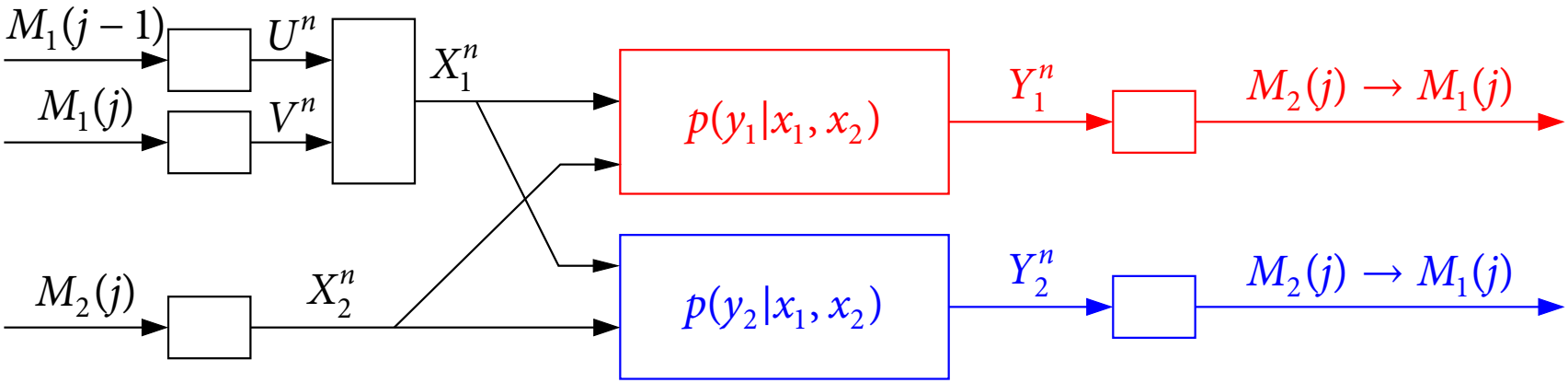


Sliding-window superposition coding (Wang et al. 2014)



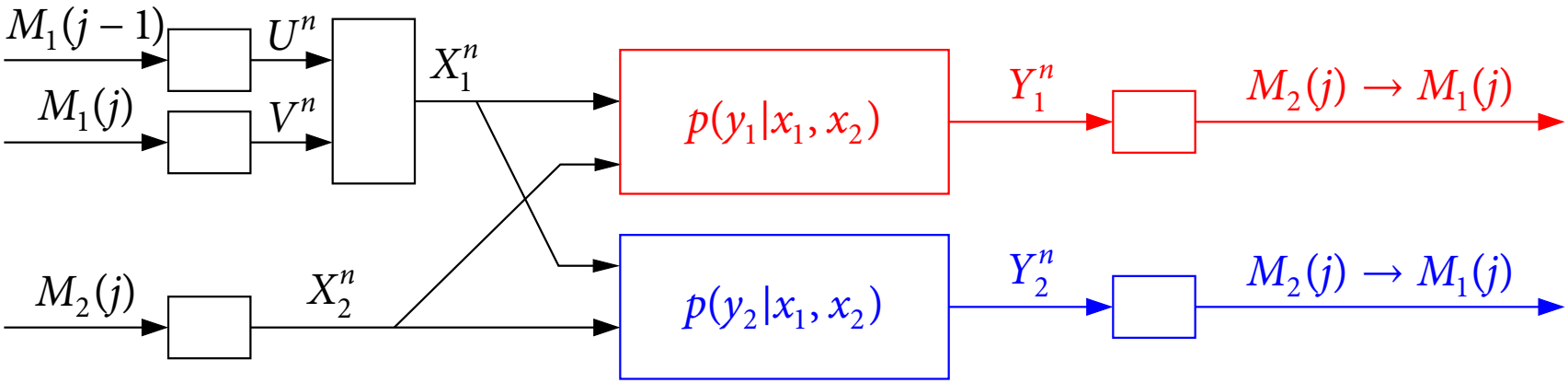
Block	1	2	3	4	5	6	7
U							
V							
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



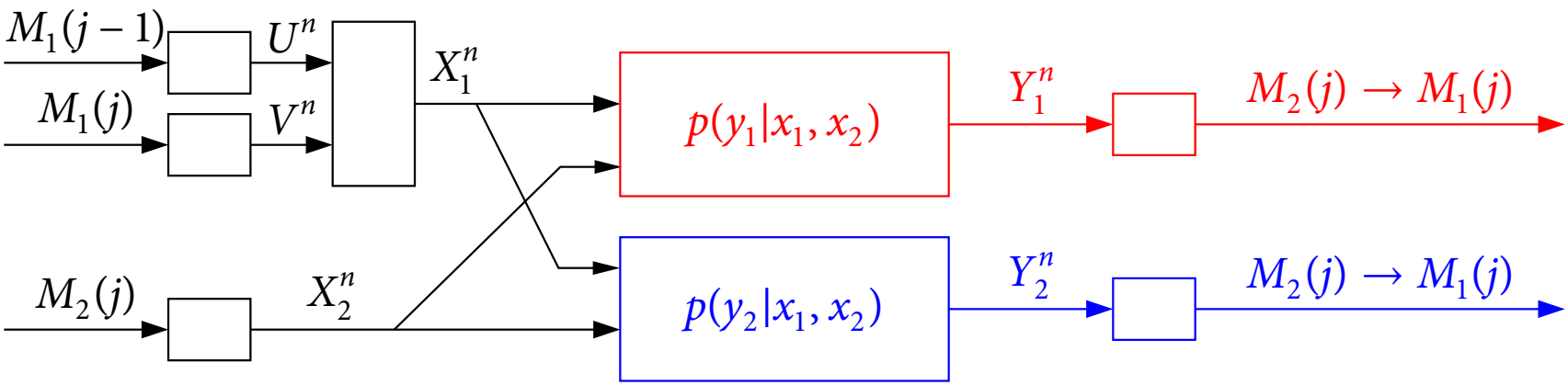
Block	1	2	3	4	5	6	7
U		$M_1(1)$					
V	$M_1(1)$						
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



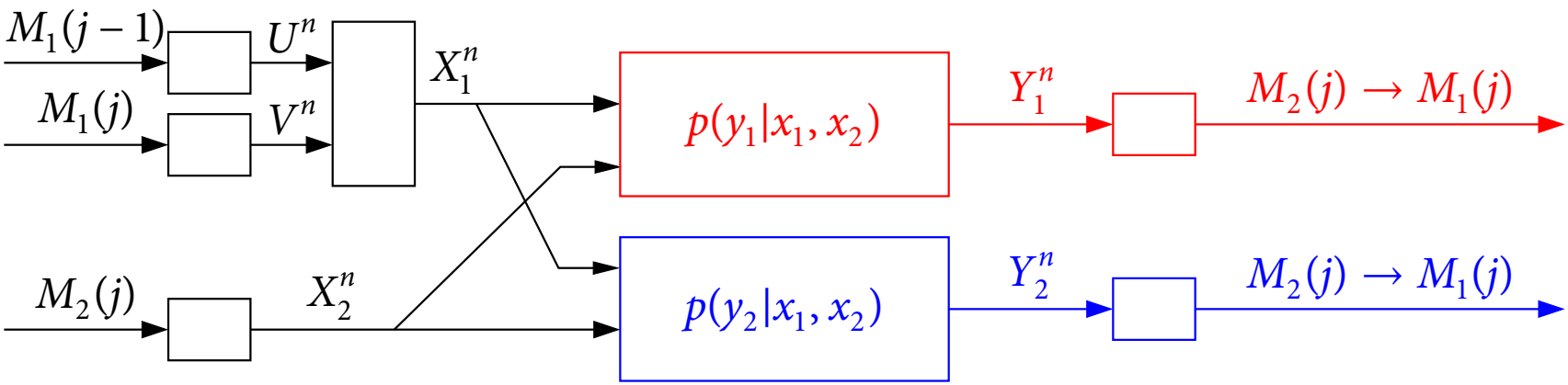
Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$				
V	$M_1(1)$	$M_1(2)$					
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



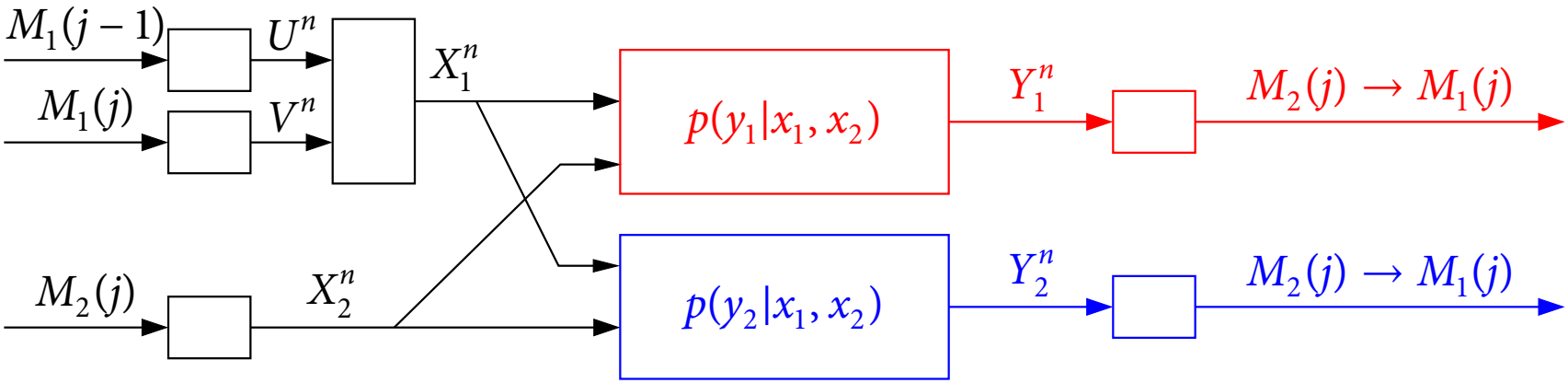
Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$			
V	$M_1(1)$	$M_1(2)$	$M_1(3)$				
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



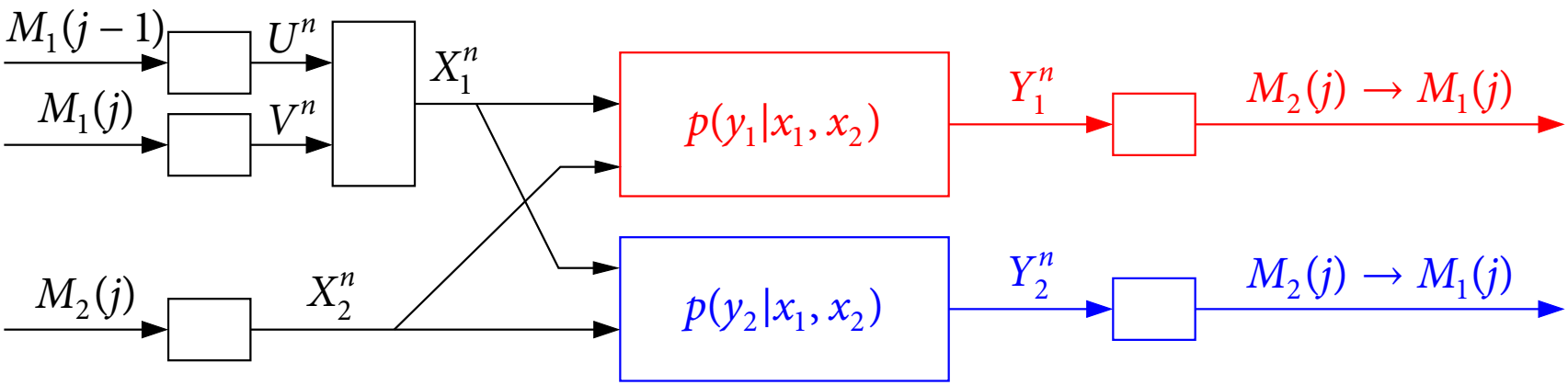
Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$		
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$			
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



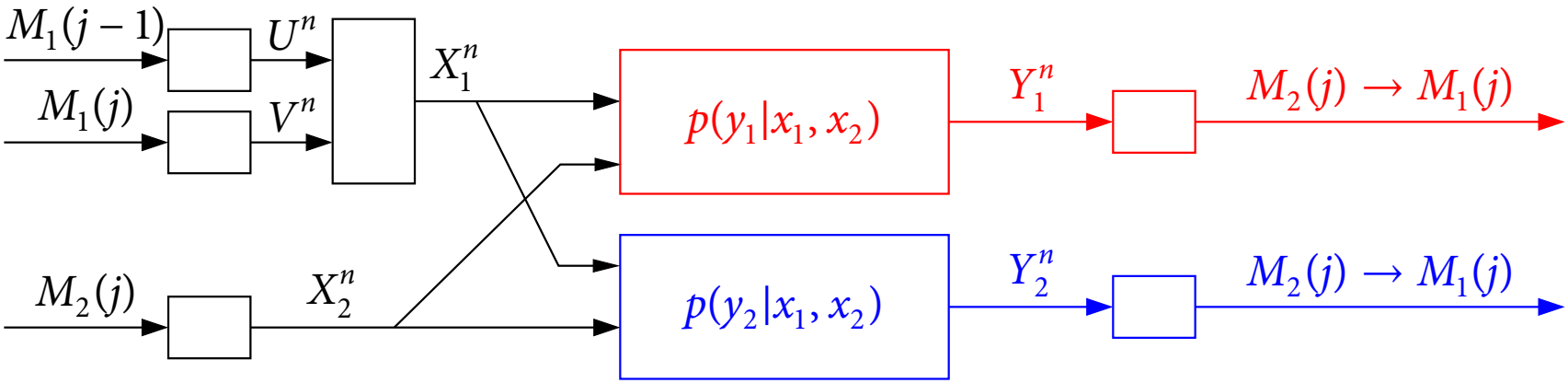
Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$		
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

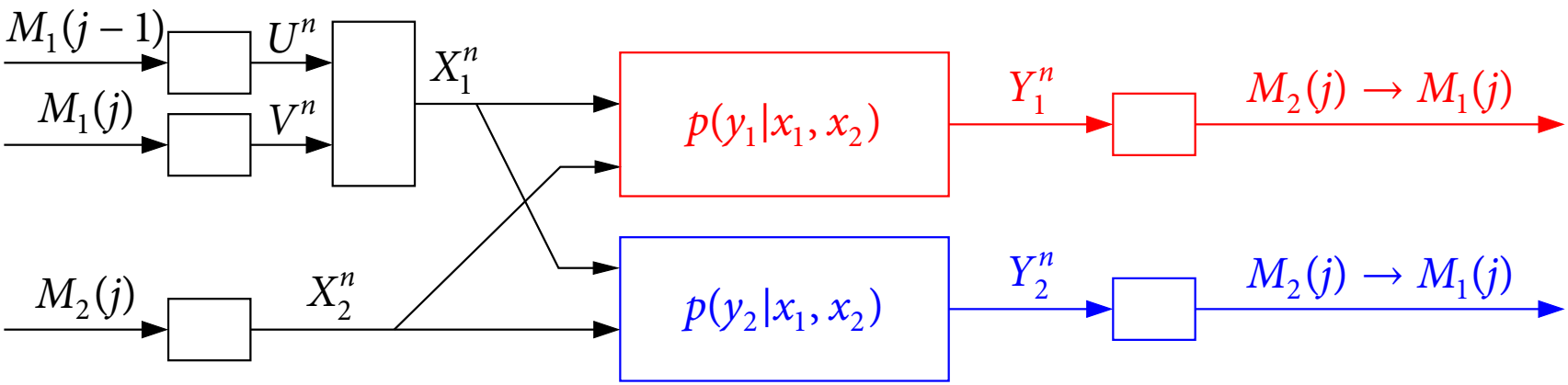
Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Block Markov coding: As in relaying and feedback communication

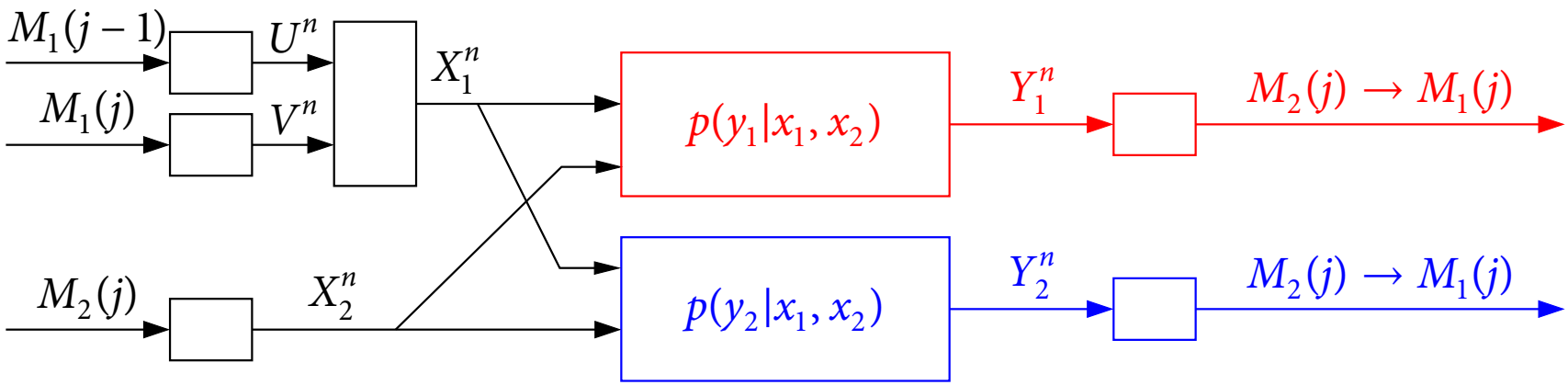
Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Block Markov coding: As in relaying and feedback communication
- Superposition coding: But without rate splitting

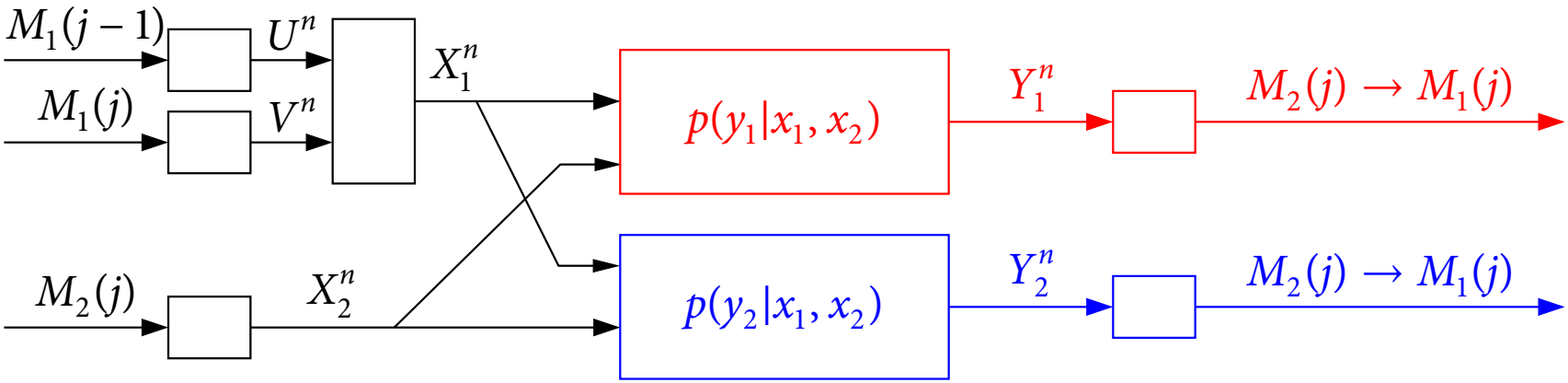
Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Block Markov coding: As in relaying and feedback communication
- Superposition coding: But without rate splitting
- Staggered (asynchronous) transmission: cf. EV-DO rev A, D-BLAST

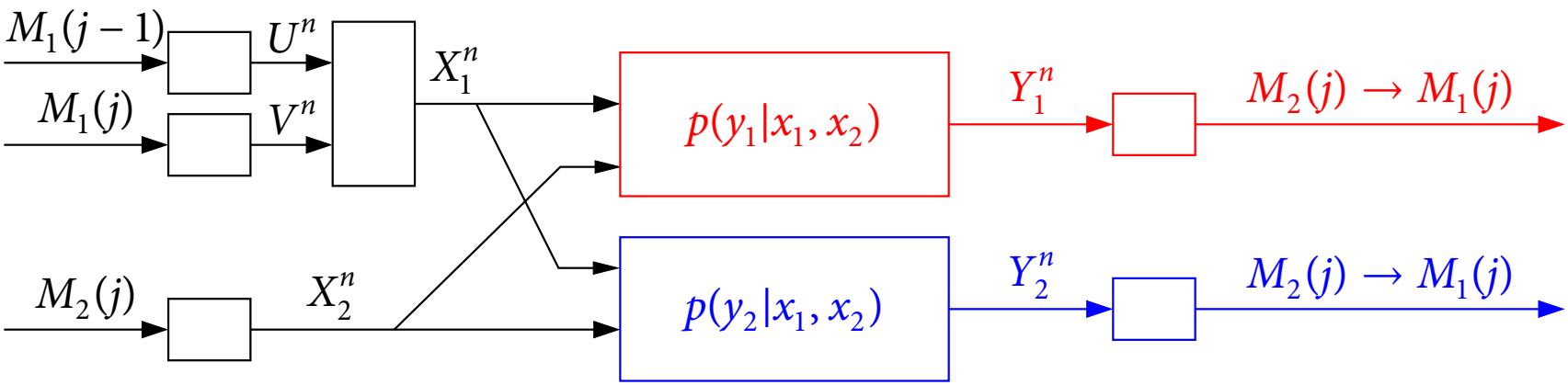
Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Sliding-window decoding

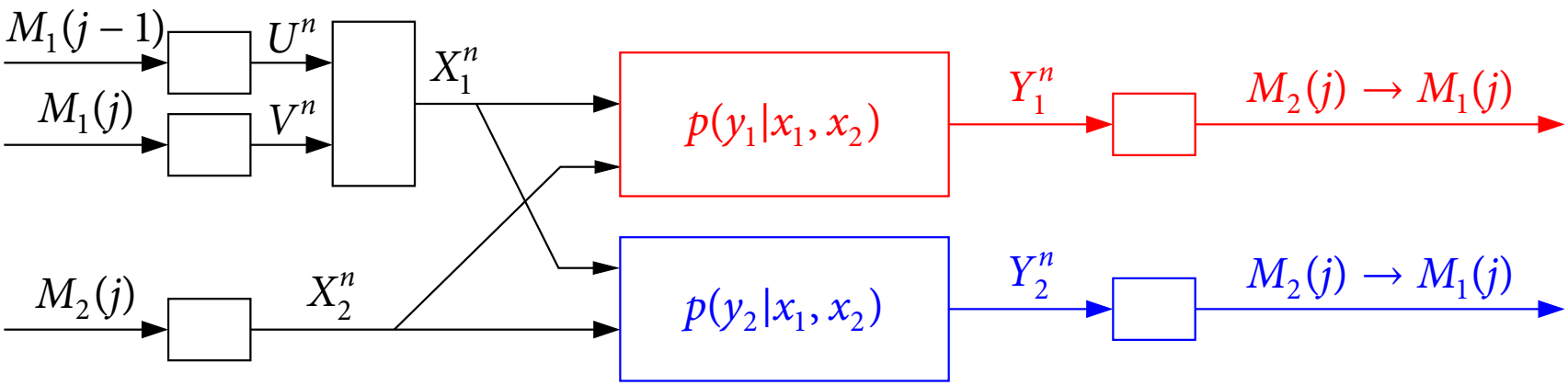
Sliding-window superposition coding (Wang et al. 2014)



Block	1	2	3	4	5	6	7
U			$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V		$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2		$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Sliding-window decoding
- Successive cancellation decoding

Sliding-window superposition coding (Wang et al. 2014)

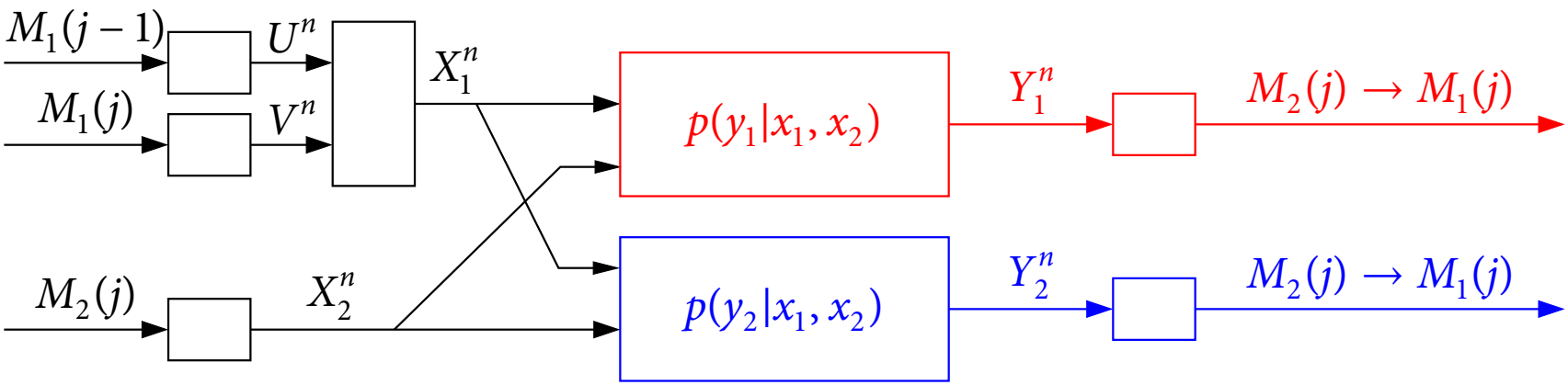


Block	1	2	3	4	5	6	7
U				$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V			$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2			$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Sliding-window decoding
- Successive cancellation decoding

$$R_2 < I(X_2; Y_j | U),$$

Sliding-window superposition coding (Wang et al. 2014)

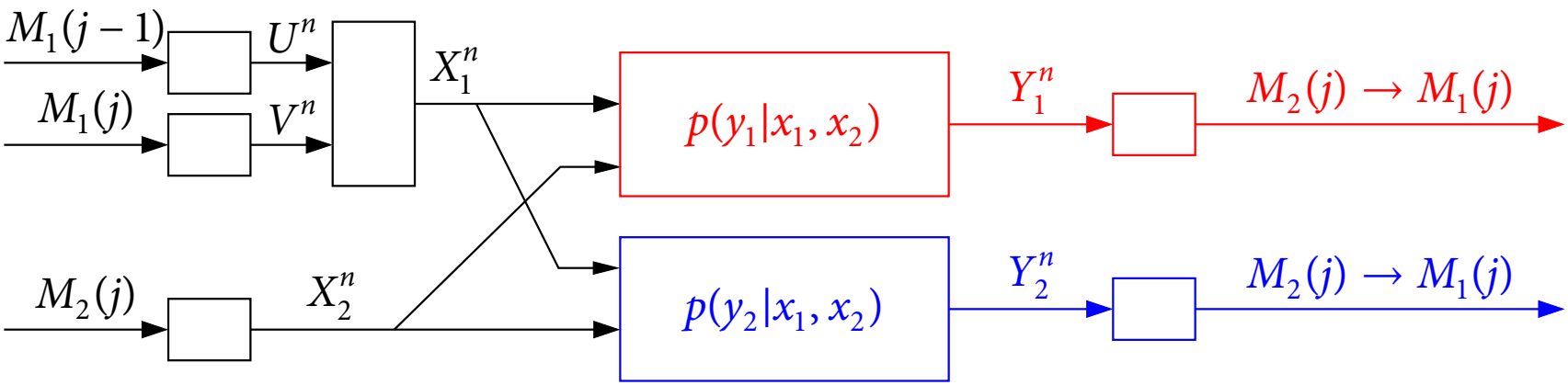


Block	1	2	3	4	5	6	7
U				$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V			$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2				$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Sliding-window decoding
- Successive cancellation decoding

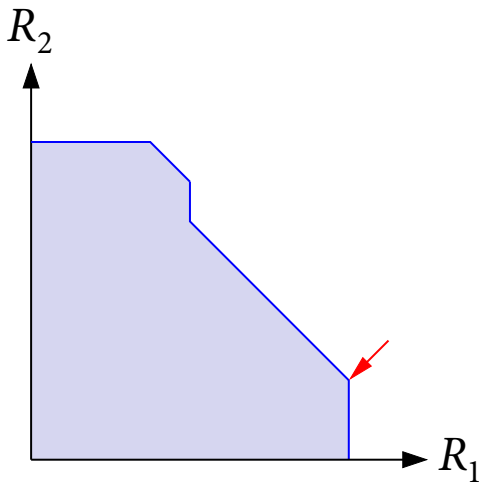
$$R_2 < I(X_2; Y_j|U),$$
$$R_1 < I(U; Y_j) + I(V; Y_j|U, X_2)$$

Sliding-window superposition coding (Wang et al. 2014)

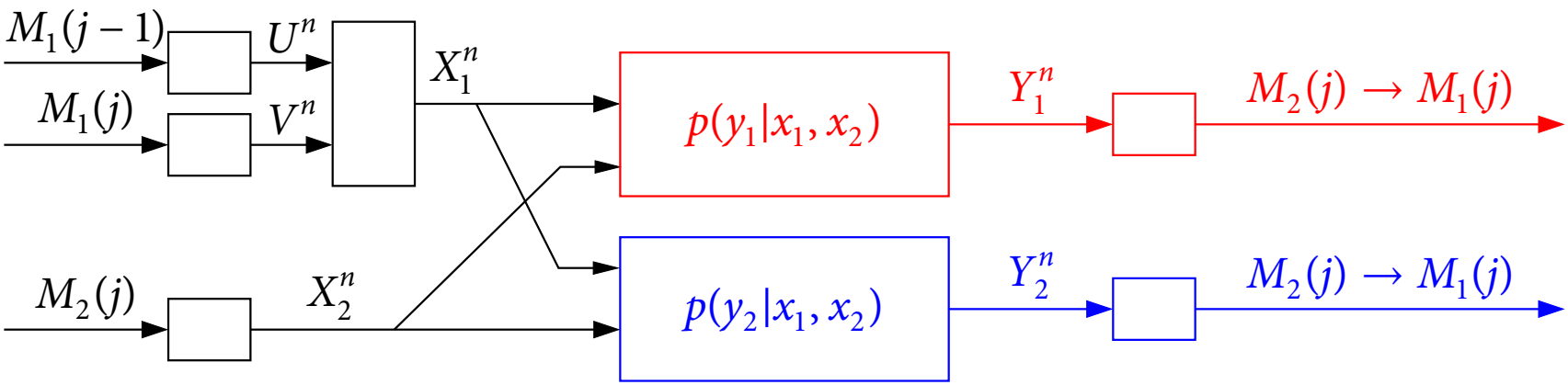


Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Every corner point: different decoding orders

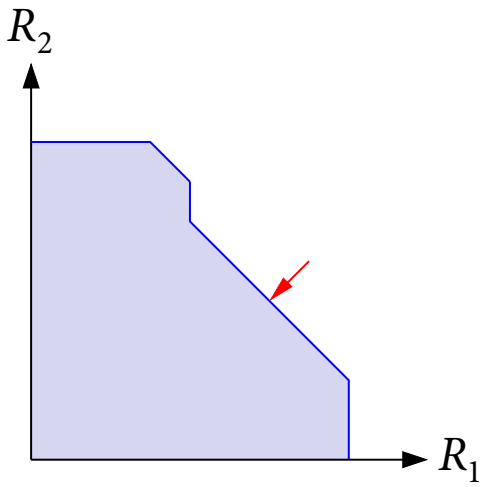


Sliding-window superposition coding (Wang et al. 2014)

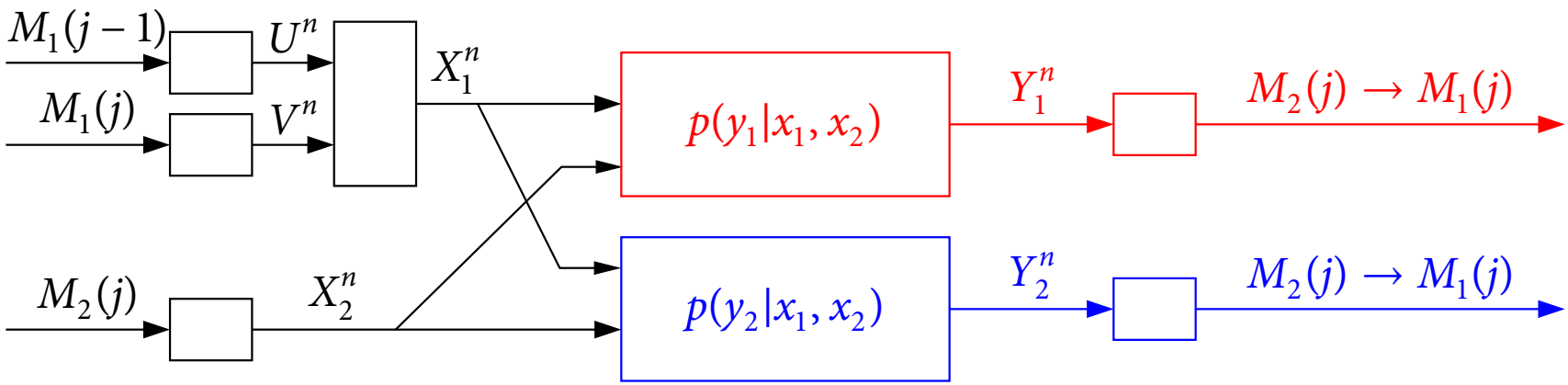


Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Every corner point: different decoding orders
- Every point: time sharing or more superposition layers

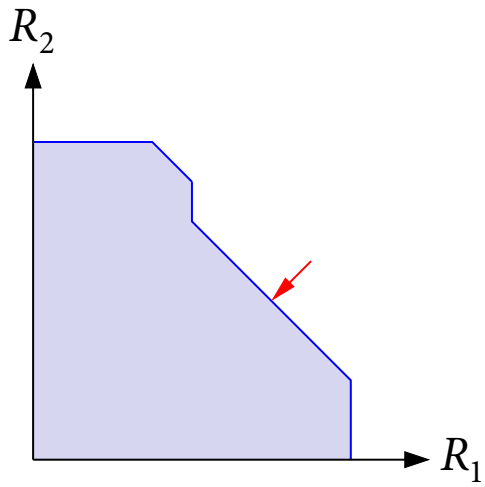


Sliding-window superposition coding (Wang et al. 2014)

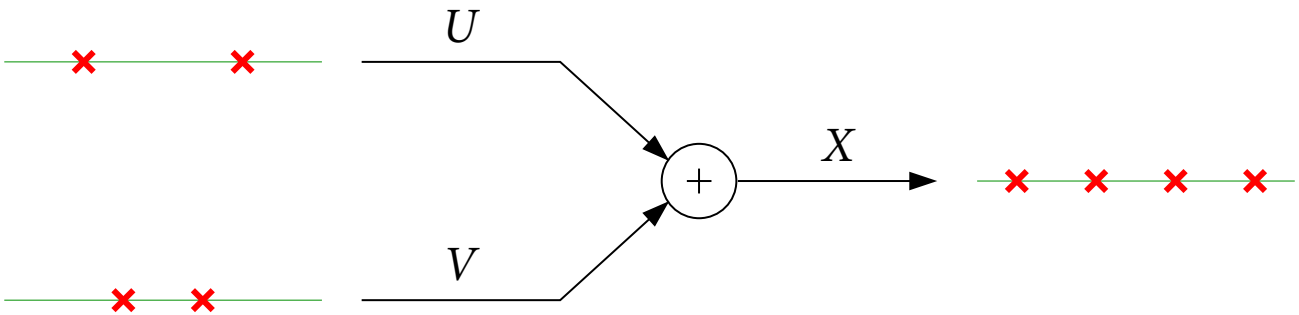


Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

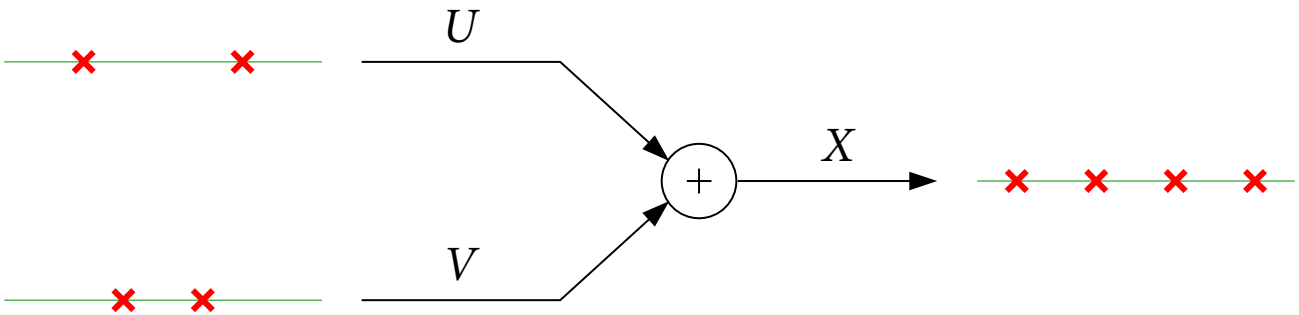
- Every corner point: different decoding orders
- Every point: time sharing or more superposition layers
- General theory for arbitrary number of users (Wang 2015)

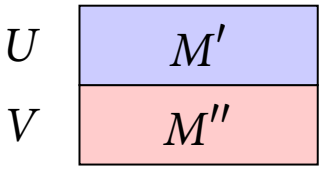


Towards coded modulation



Towards coded modulation



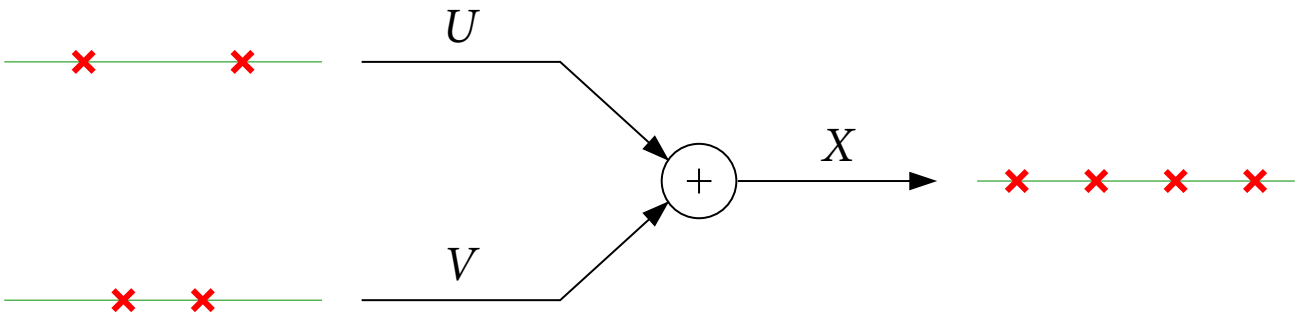


Multilevel coding (MLC)

$$R' < I(U;Y)$$
$$R'' < I(V;Y|U)$$

Short, nonuniversal

Towards coded modulation



Multilevel coding (MLC)

Bit-interleaved coded modulation (BICM)

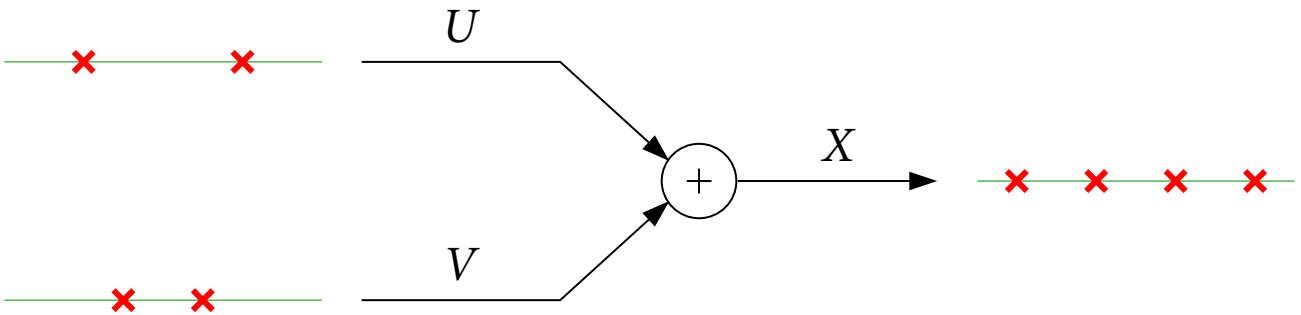
$$R' < I(U;Y)$$
$$R'' < I(V;Y|U)$$

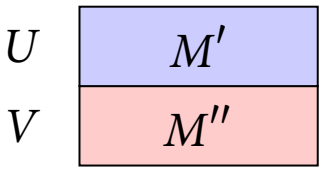
$$R < I(U;Y) + I(V;Y)$$

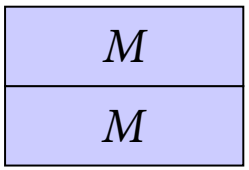
Short, nonuniversal

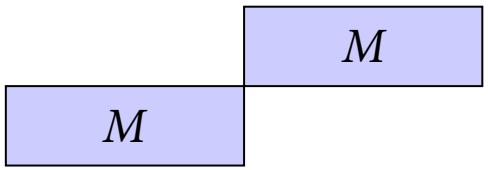
Other layers as noise

Towards coded modulation









Multilevel coding (MLC)

$$R' < I(U;Y)$$
$$R'' < I(V;Y|U)$$

Short, nonuniversal

Bit-interleaved coded modulation (BICM)

$$R < I(U;Y) + I(V;Y)$$

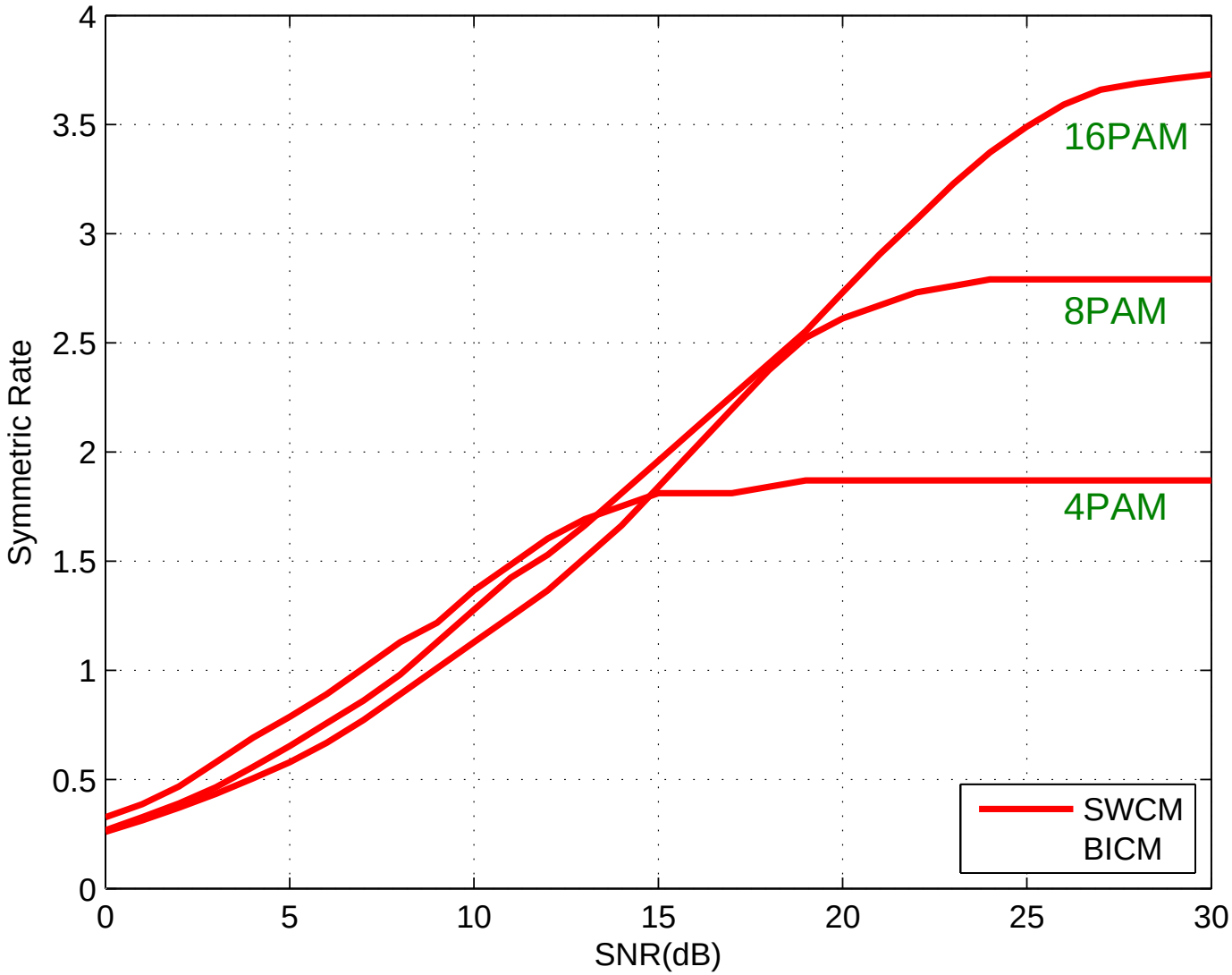
Other layers as noise

Sliding-window coded modulation (SWCM)

$$R < I(U;Y) + I(V;Y|U)$$
$$= I(X;Y)$$

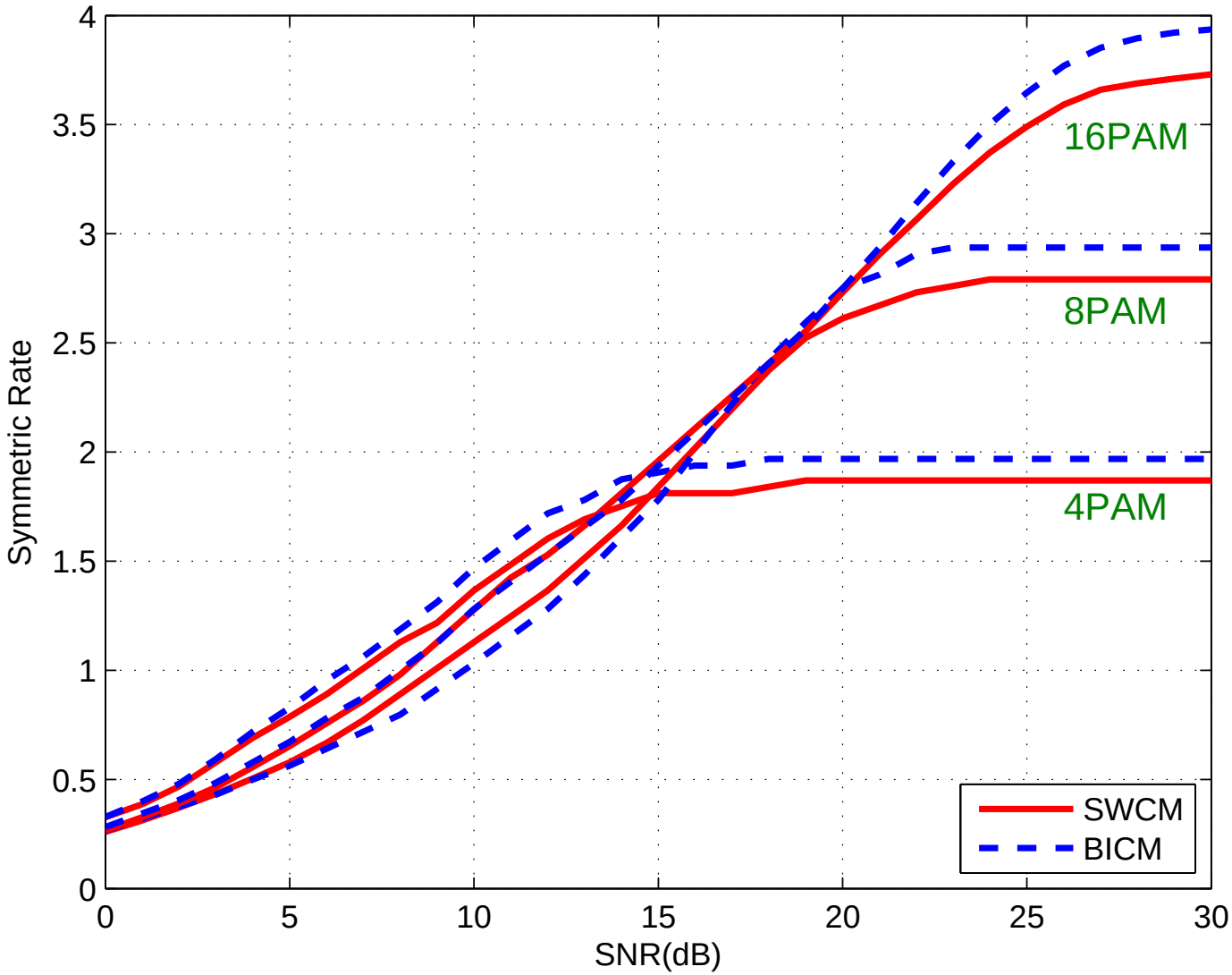
Error prop., rate loss

P2P performance



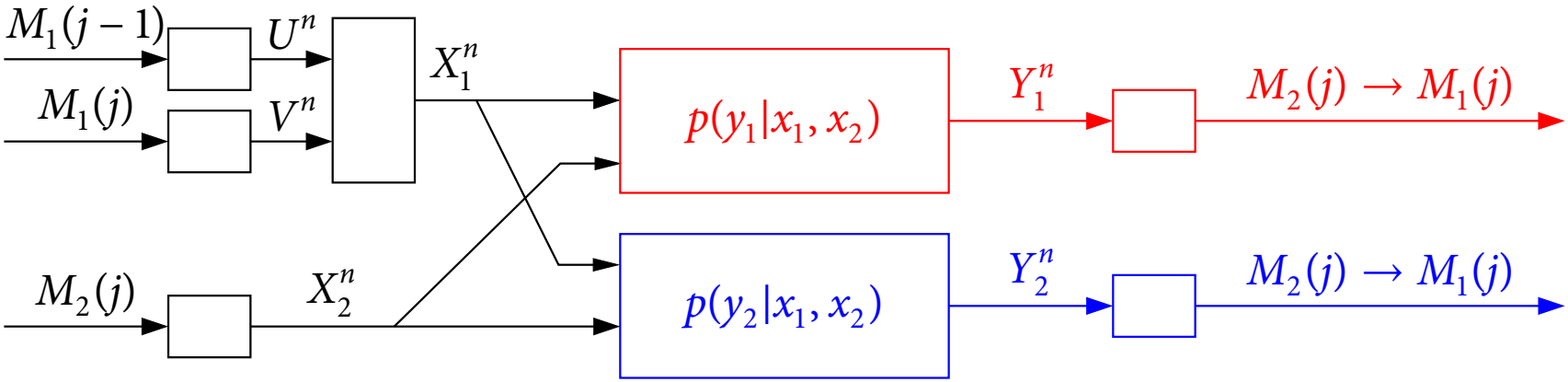
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 2048$, BLER $= 0.1$

P2P performance



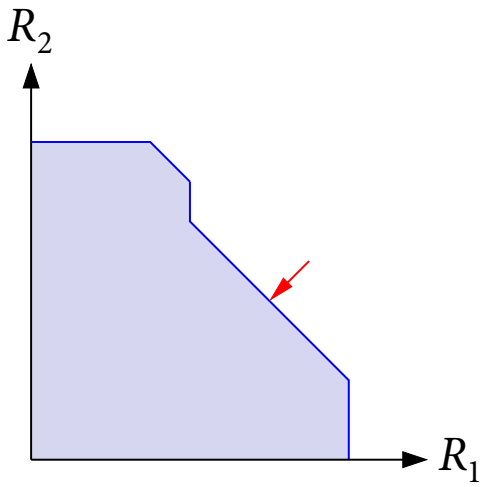
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 2048$, BLER $= 0.1$

Back to interference mitigation

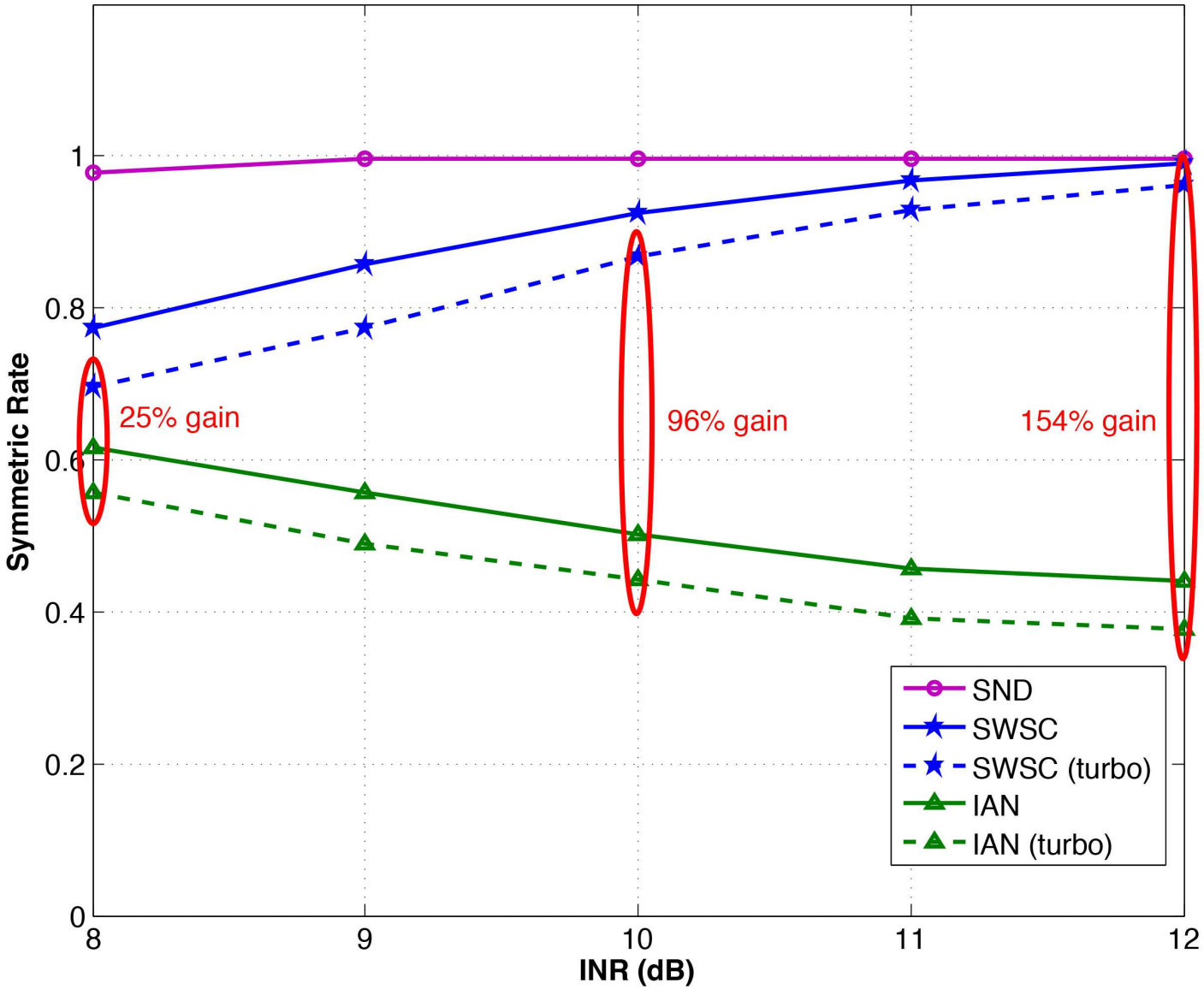


Block	1	2	3	4	5	6	7
U		$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$
V	$M_1(1)$	$M_1(2)$	$M_1(3)$	$M_1(4)$	$M_1(5)$	$M_1(6)$	
X_2	$M_2(1)$	$M_2(2)$	$M_2(3)$	$M_2(4)$	$M_2(5)$	$M_2(6)$	$M_2(7)$

- Every corner point: different decoding orders
- Every point: time sharing or more superposition layers
- General theory for arbitrary number of users (Wang 2015)

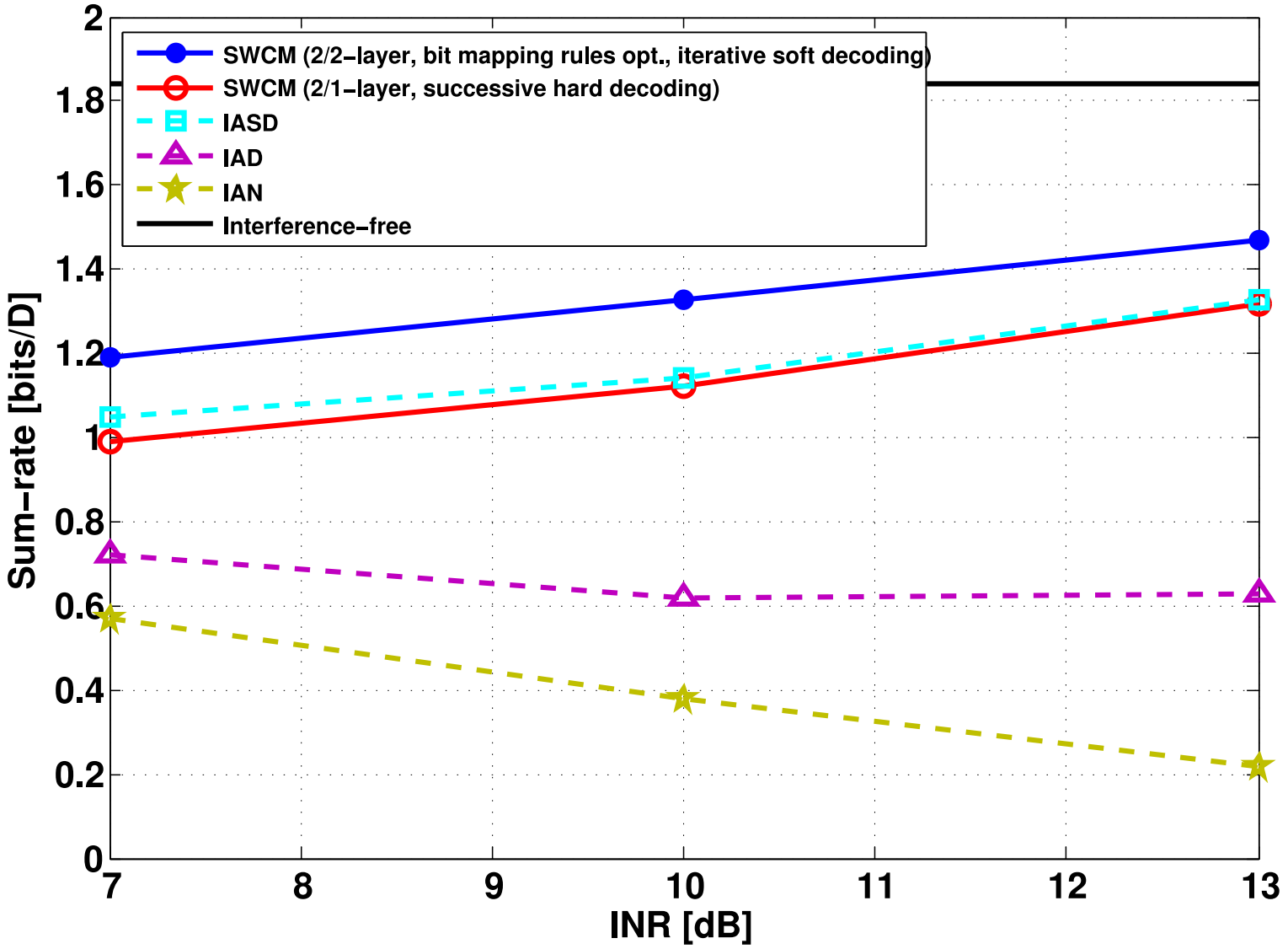


Gaussian channel performance (Park–K–Wang 2014)



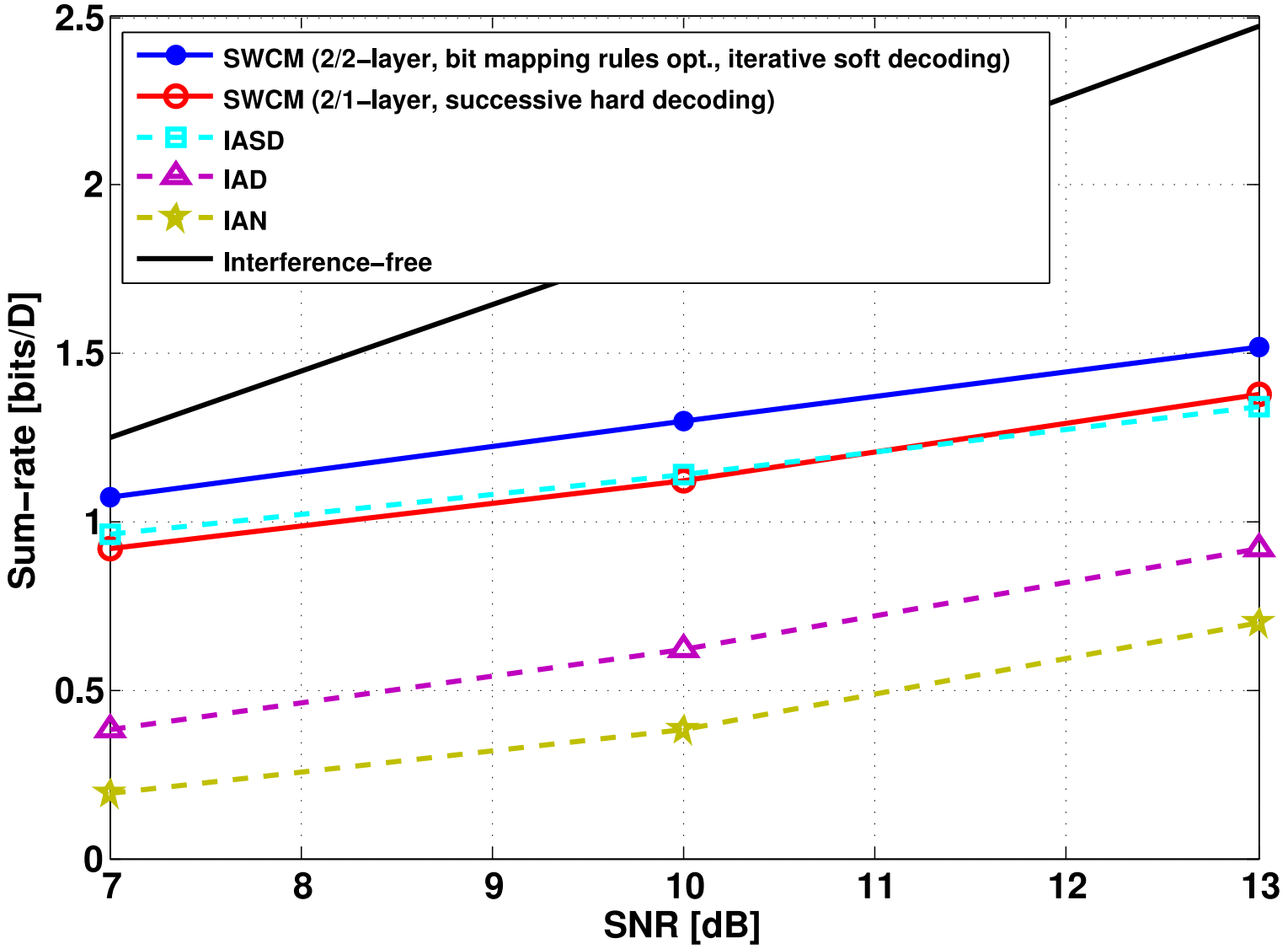
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 2048$, BLER $= 0.1$, SNR $= 10$ dB

OFDM/Ped-B channel performance (Kim et al. 2015)



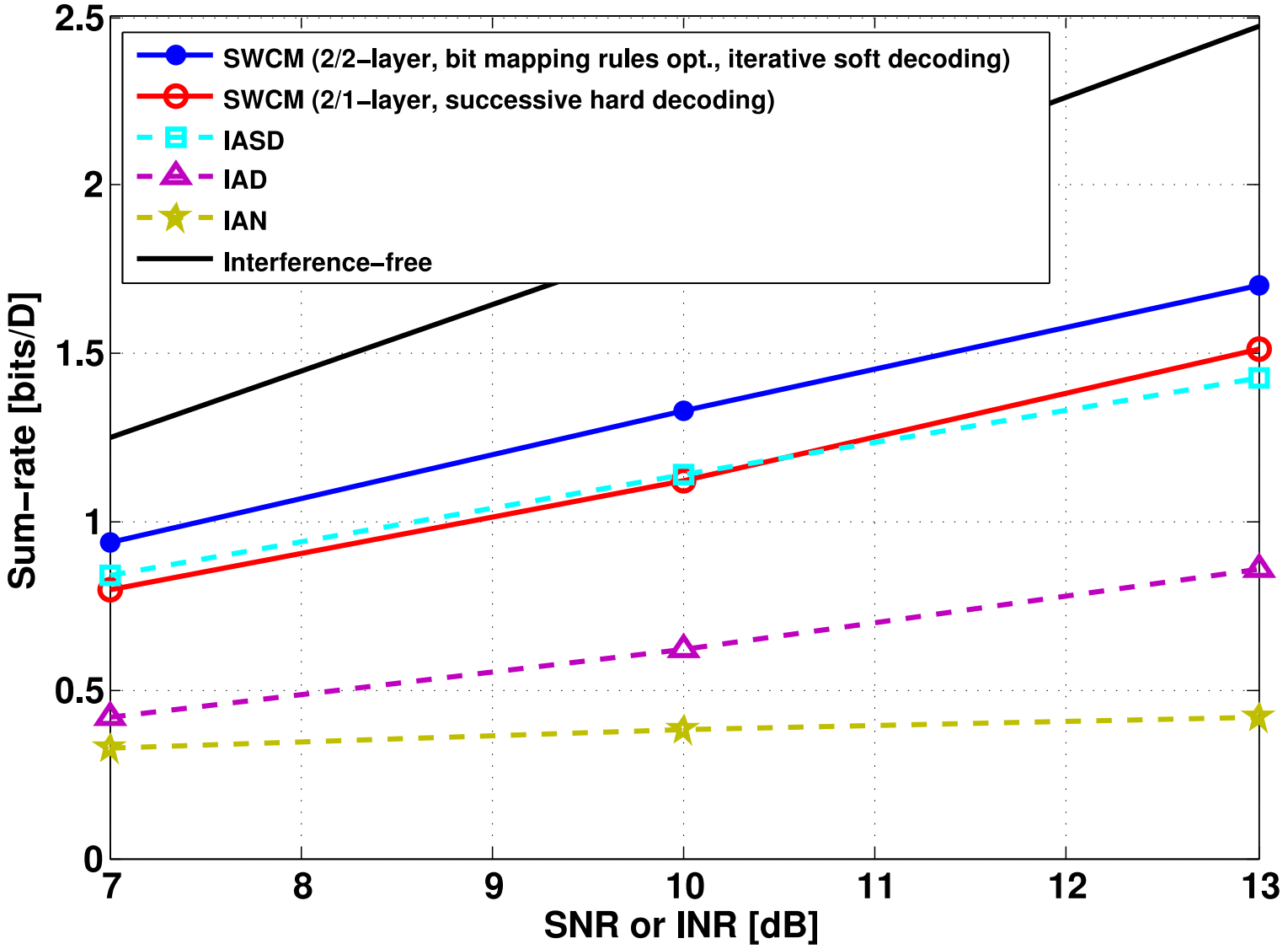
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER $= 0.1$

OFDM/Ped-B channel performance (Kim et al. 2015)



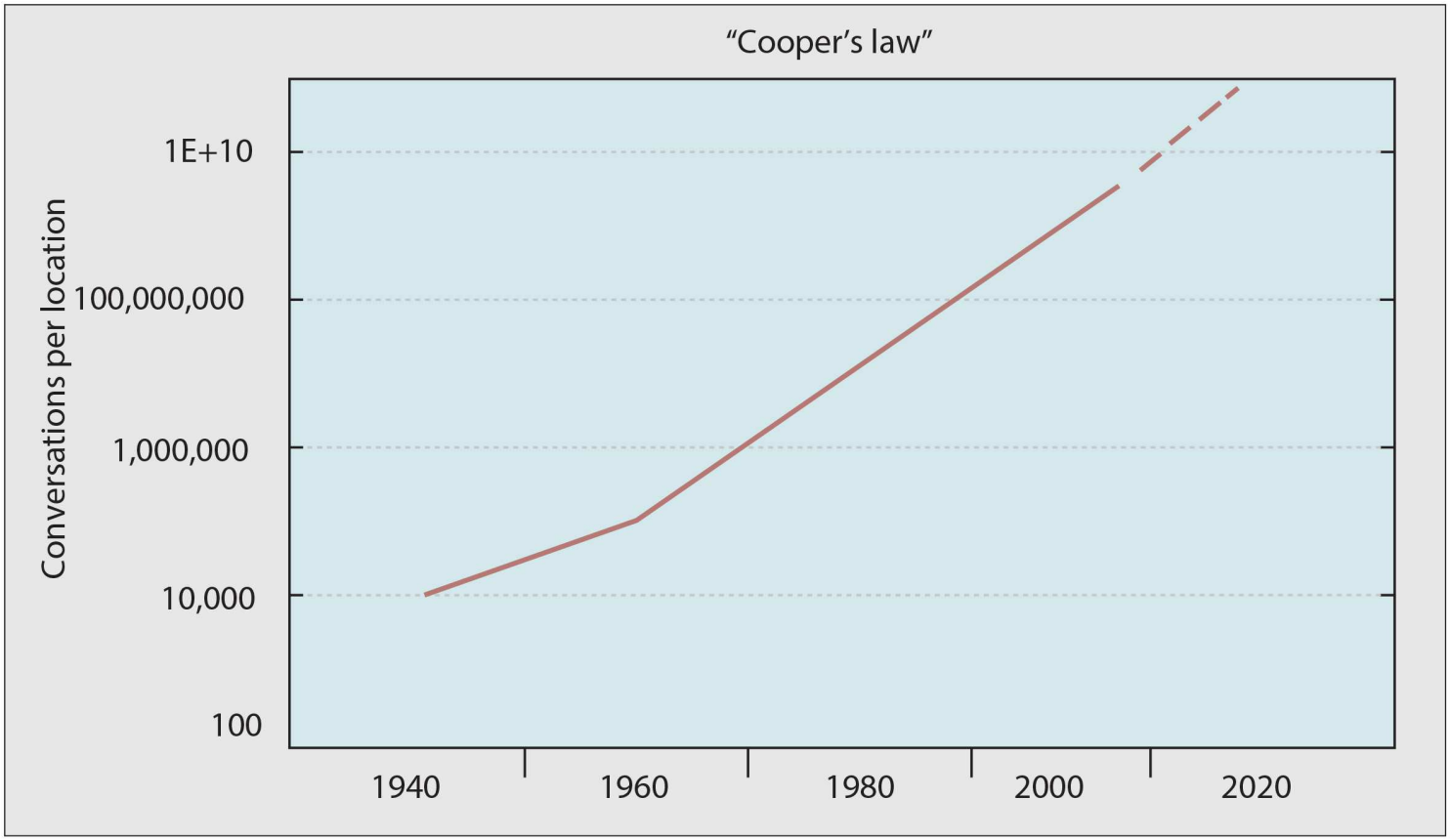
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER $= 0.1$

OFDM/Ped-B channel performance (Kim et al. 2015)



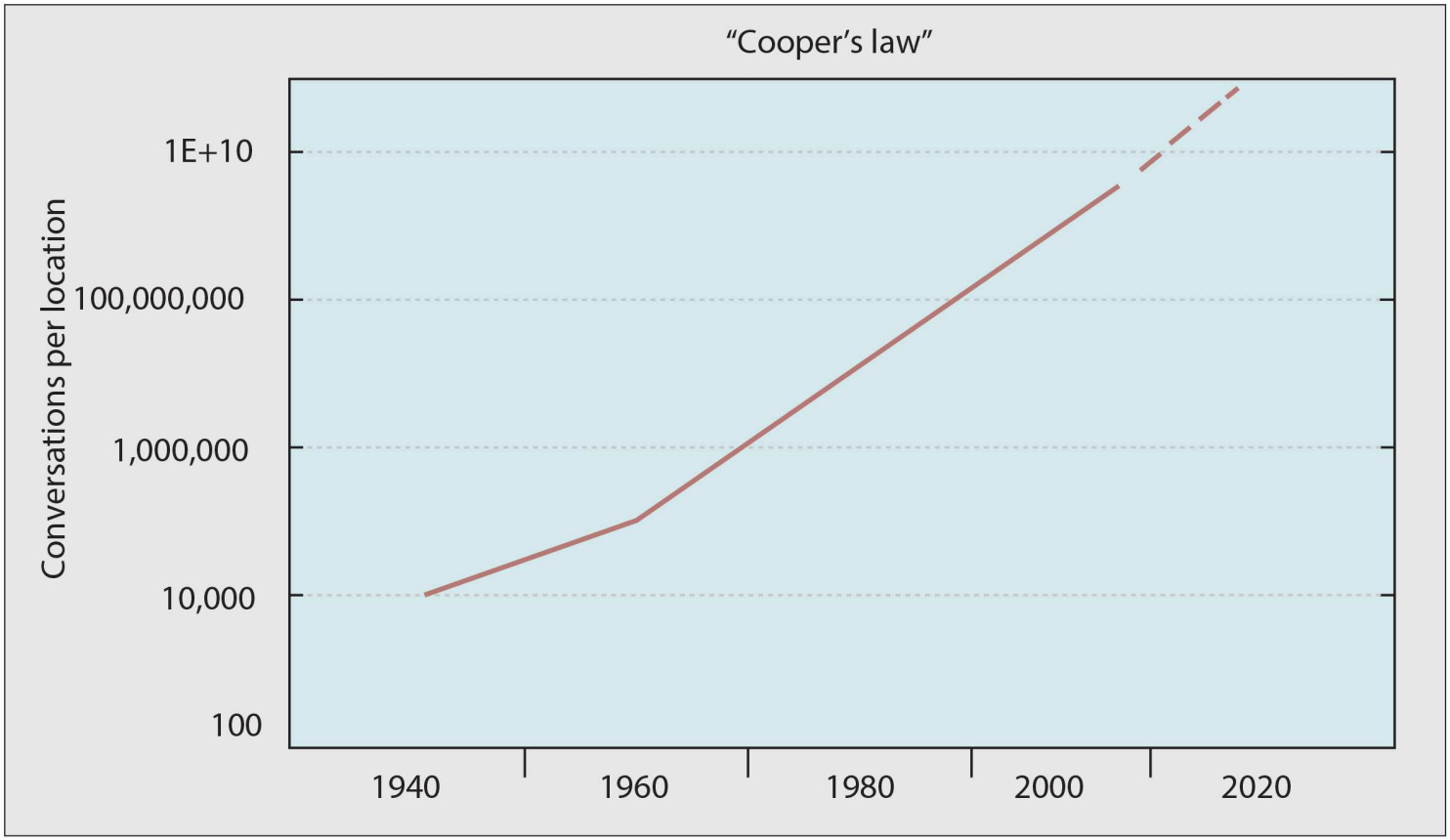
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER $= 0.1$

Cooper's Law



Source: Arraycomm, Zander–Mähönen (2013)

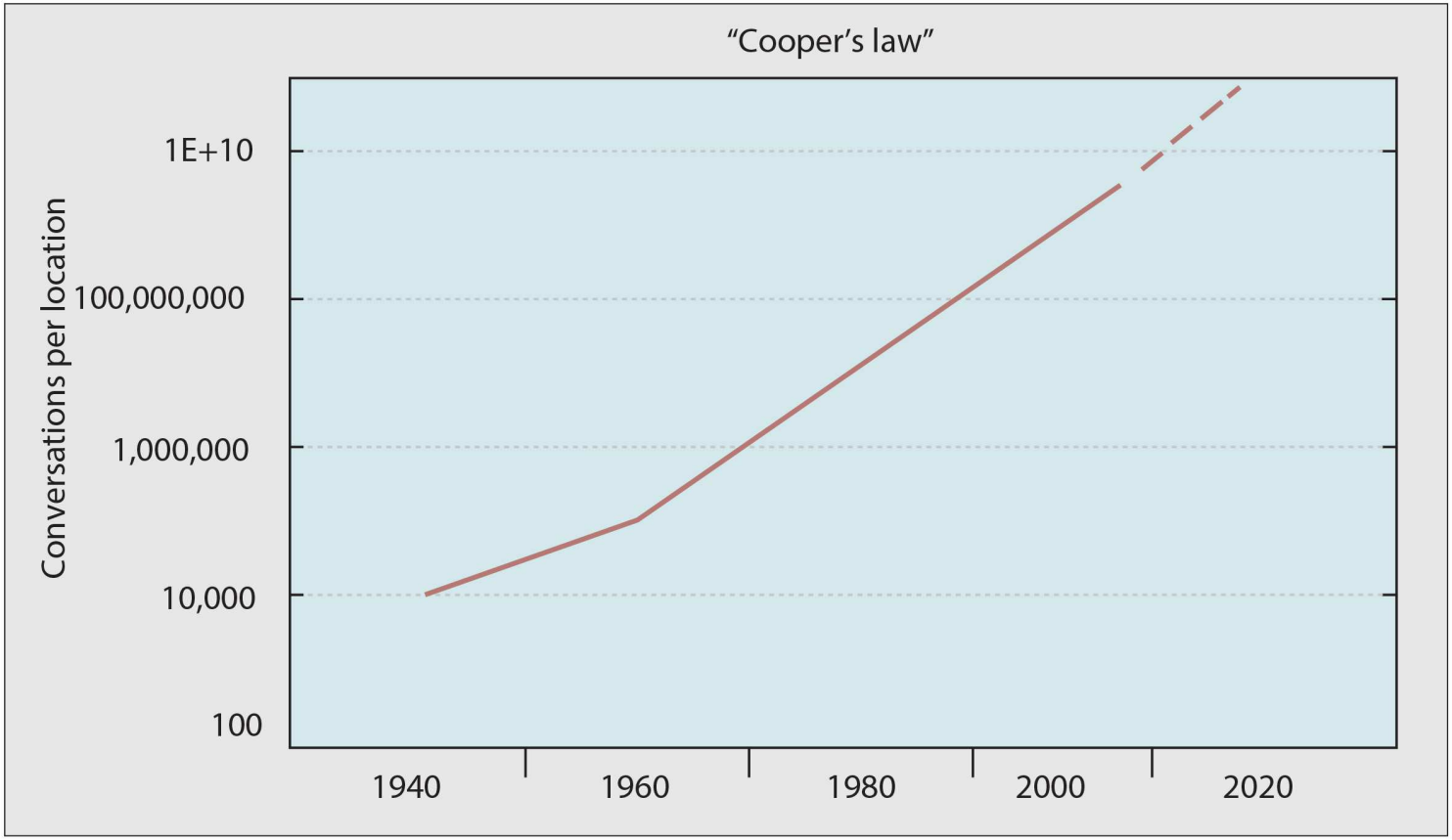
Cooper's Law



Source: Arraycomm, Zander–Mähönen (2013)

- Gain over the past 45 years = $10^6 \propto \eta W_{\text{sys}} N_{\text{BS}}$

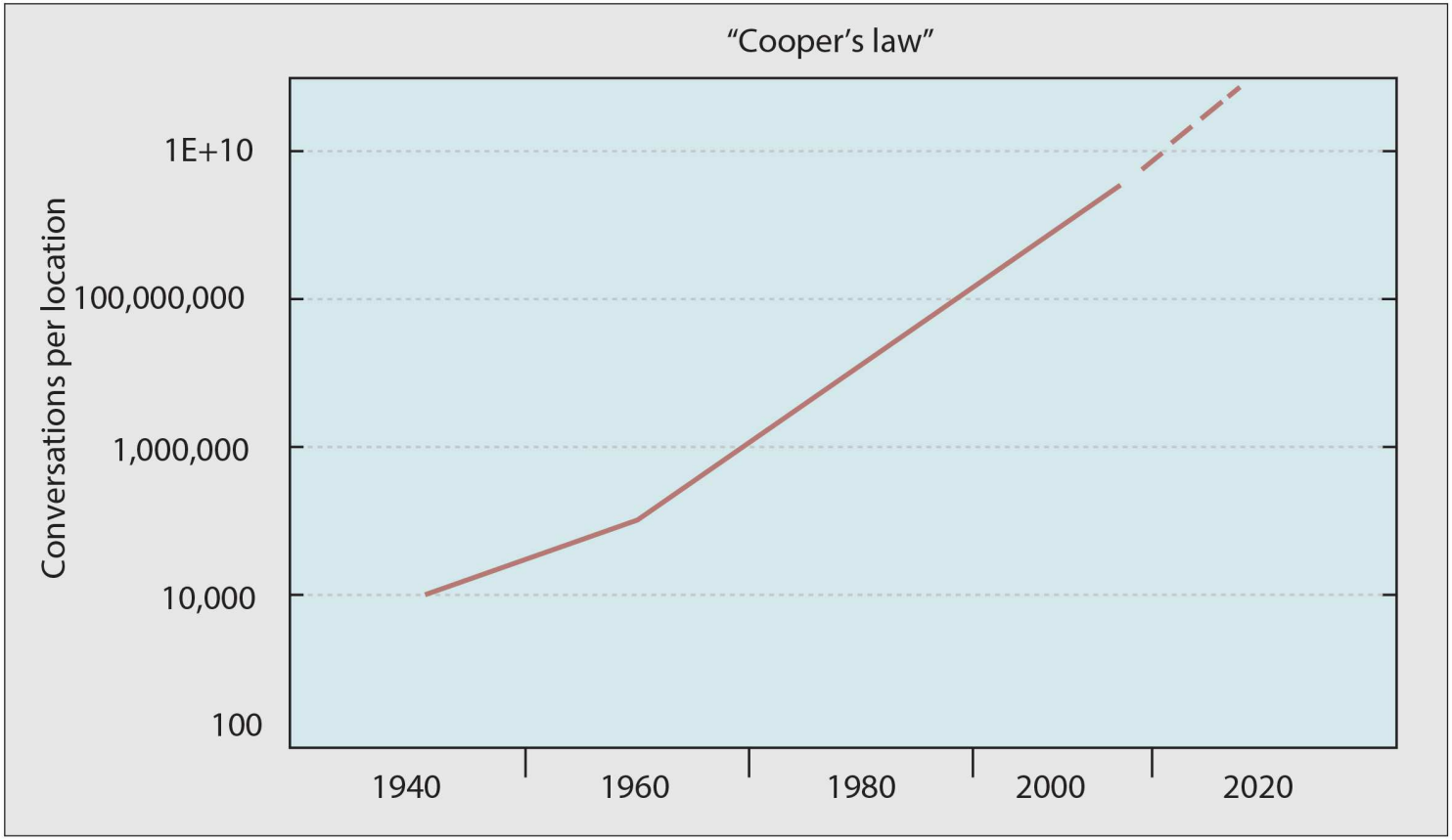
Cooper's Law



Source: Arraycomm, Zander–Mähönen (2013)

- Gain over the past 45 years = $10^6 \propto \eta W_{\text{sys}} N_{\text{BS}}$
  - Spectral efficiency η: x 25

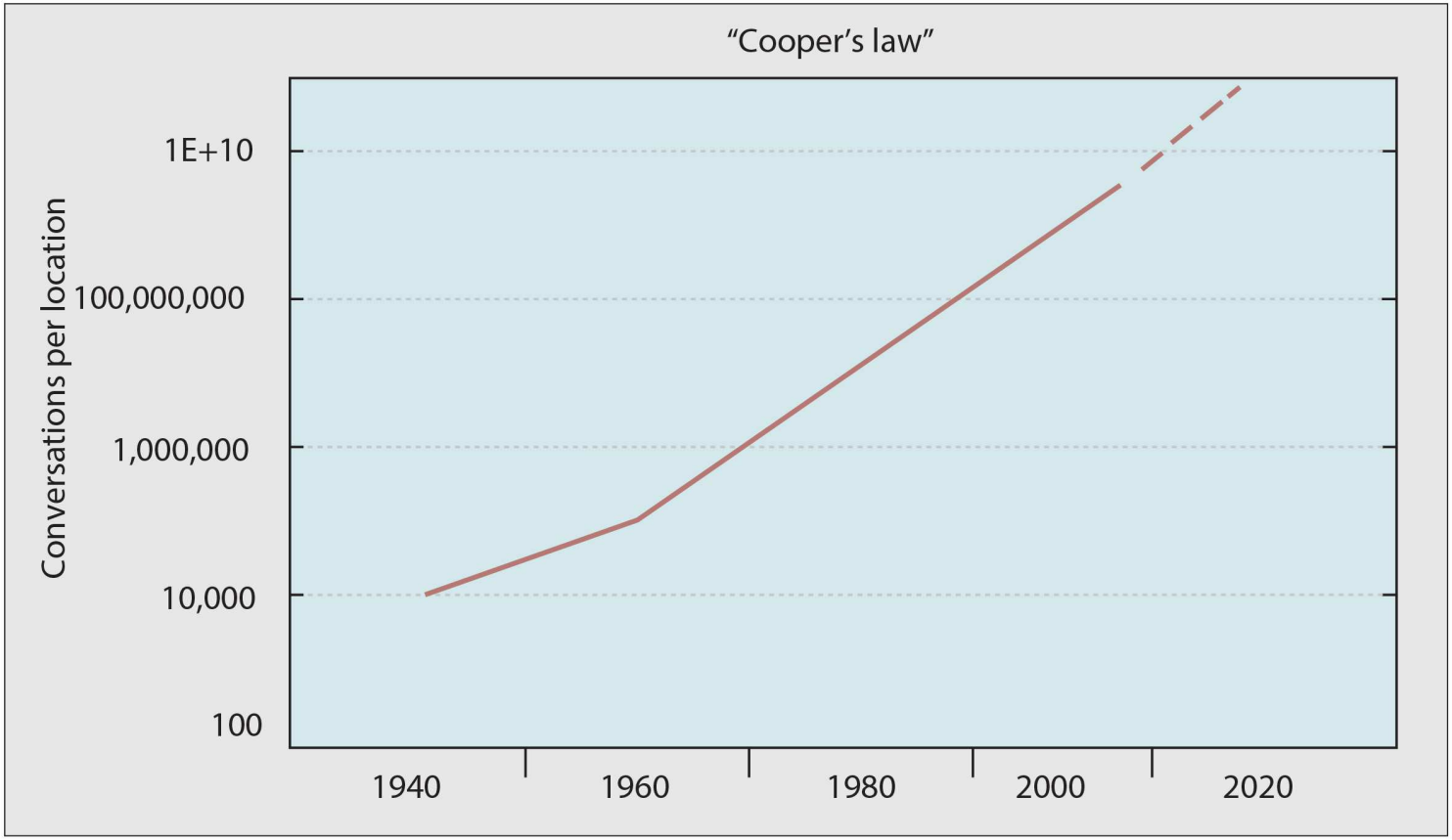
Cooper's Law



Source: Arraycomm, Zander–Mähönen (2013)

- Gain over the past 45 years $= 10^6 \propto \eta W_{\text{sys}} N_{\text{BS}}$
  - Spectral efficiency η: x 25
  - System bandwidth W_{sys}: x 25

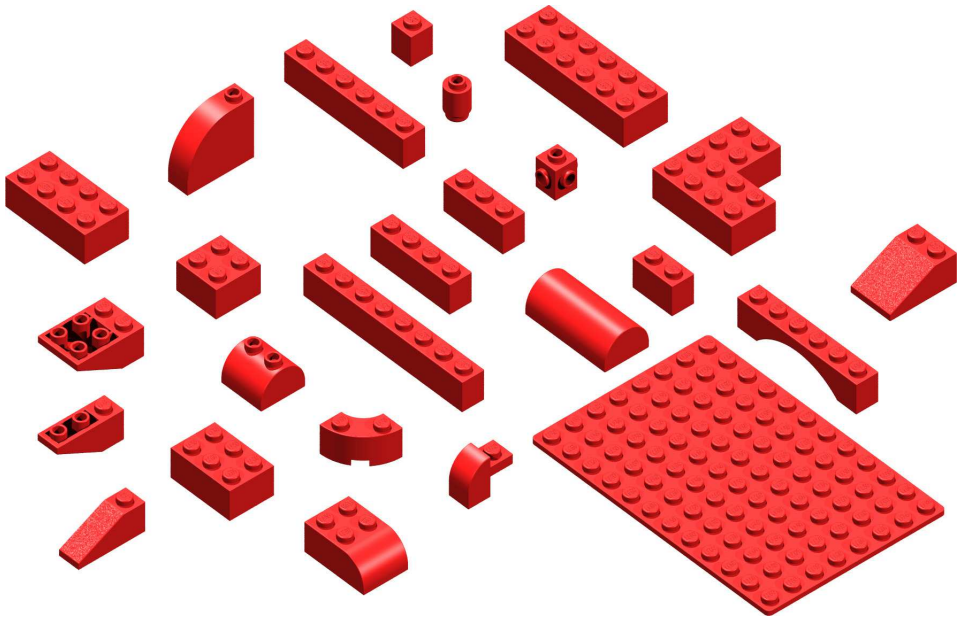
Cooper's Law



Source: Arraycomm, Zander–Mähönen (2013)

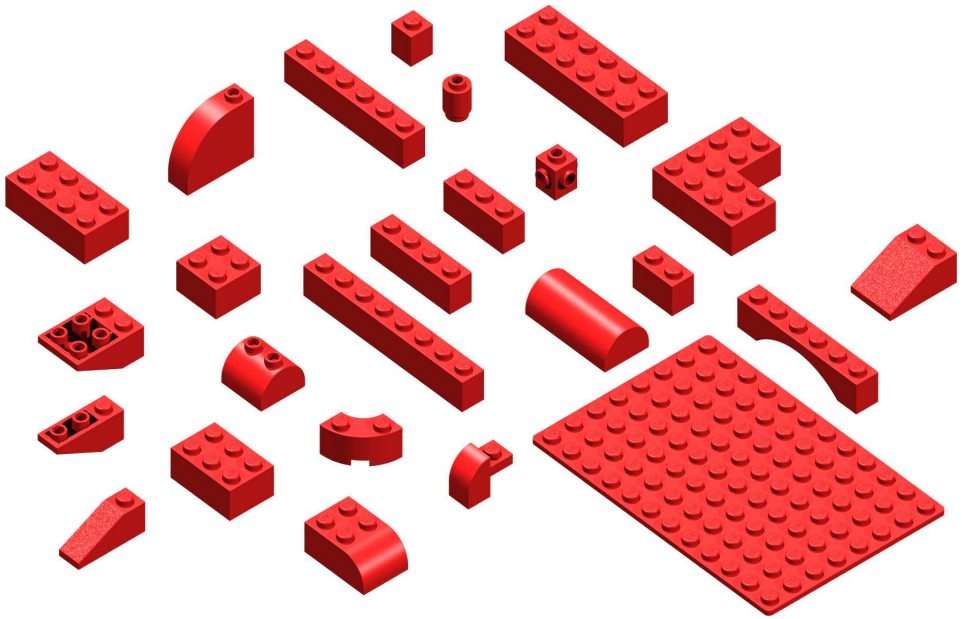
- Gain over the past 45 years $= 10^6 \propto \eta W_{\text{sys}} N_{\text{BS}}$
  - Spectral efficiency η: x 25
  - System bandwidth W_{sys}: x 25
  - # of base stations N_{BS}: x 1600 (spatial reuse of frequency)

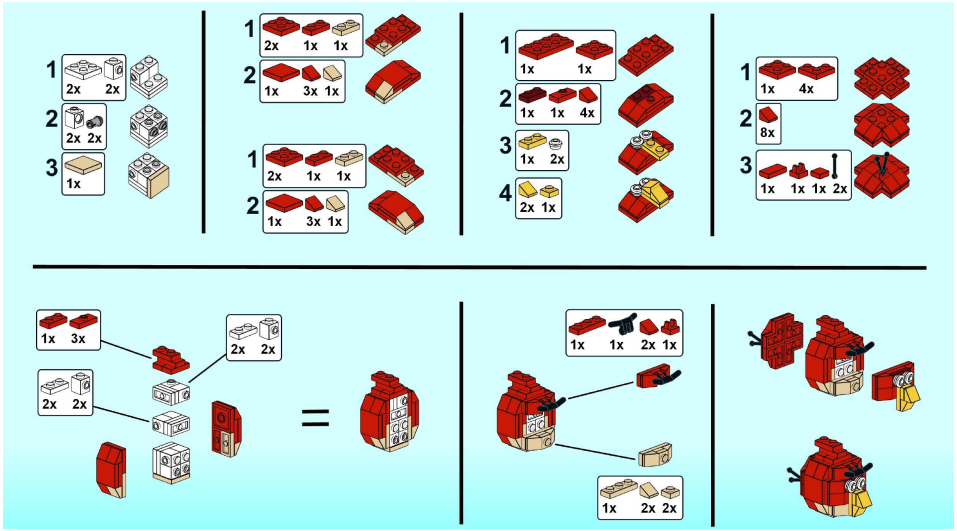
What's next?



- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)

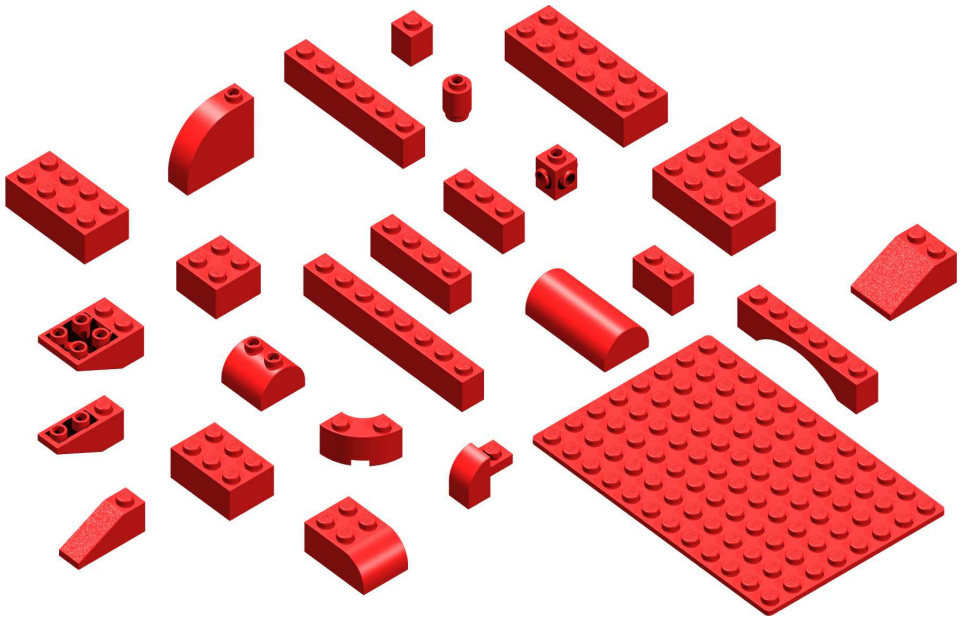
What's next?

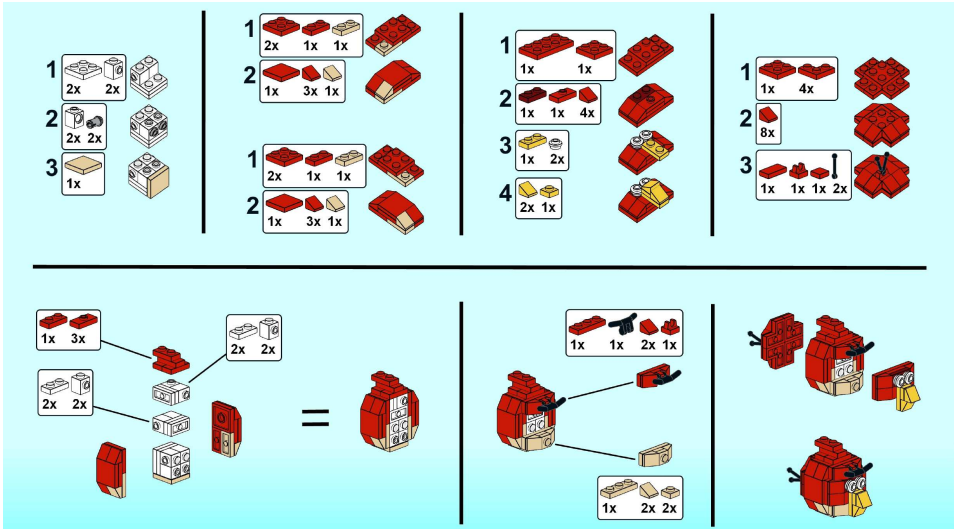




- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?





- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?

- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?

- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?



- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?




- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding

What's next?

Let's have fun building better networks!

- Point-to-point codes (random coding)
- Superposition coding
- Successive cancellation decoding
- Simultaneous decoding
- Multicoding (writing on dirty paper)
- Random binning (Slepian–Wolf)
- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding