Bidirectional Multi-Hop Communication via Two Relays Using Nested Voronoi Codes

Christian Häger[†] Aydin Sezgin[‡] Alexandre Graell i Amat[†]

[†]Department of Signals and Systems
Communication Systems Group
Chalmers University of Technology
Gothenburg, Sweden

[‡]Chair of Communication Systems
Ruhr University Bochum
Bochum, Germany

{christian.haeger, alexandre.graell}@chalmers.se, aydin@cs.rub.de

March 14, 2012

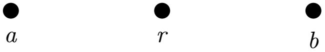
CHALMERS

Outline

1. Basic Principles
2. One Relay (Separated Two-Way Relay Channel)
3. Two Relays (Separated Two-Way Two-Relay Channel)
4. Conclusion

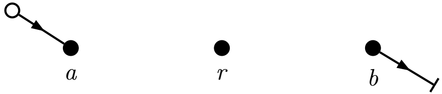
Message Exchange via a Relay

Message Exchange via a Relay



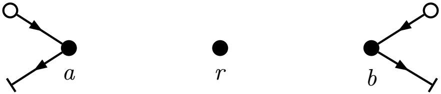
- Three nodes / devices

Message Exchange via a Relay



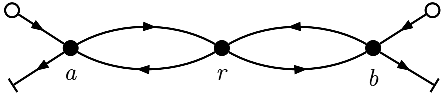
- Three nodes / devices
- One message from a to b

Message Exchange via a Relay



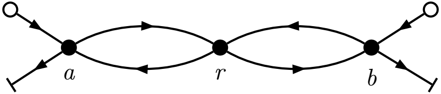
- Three nodes / devices
- One message from a to b
- One message from b to a

Message Exchange via a Relay



- Three nodes / devices
- One message from a to b
- One message from b to a
- No direct link

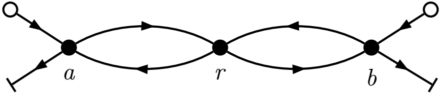
Message Exchange via a Relay



- Three nodes / devices
- One message from a to b
- One message from b to a
- No direct link

Start with toy problems:

Message Exchange via a Relay

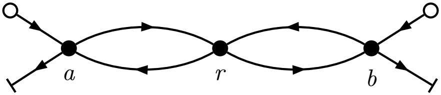


- Three nodes / devices
- One message from a to b
- One message from b to a
- No direct link

Start with toy problems:

- No noise at the nodes

Message Exchange via a Relay

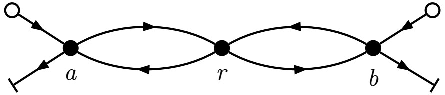


- Three nodes / devices
- One message from a to b
- One message from b to a
- No direct link

Start with toy problems:

- No noise at the nodes
- Binary input and output alphabets

Message Exchange via a Relay



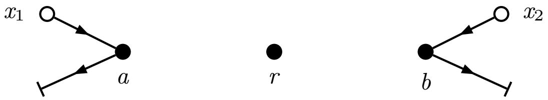
- Three nodes / devices
- One message from a to b
- One message from b to a
- No direct link

Start with toy problems:

- No noise at the nodes
- Binary input and output alphabets
- Half-duplex constraint (i.e. a node either listens or talks but not both at the same time)

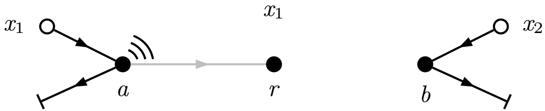
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



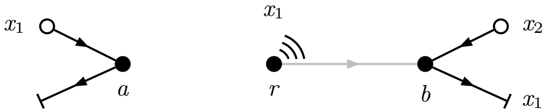
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



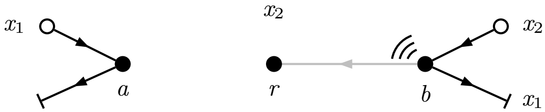
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



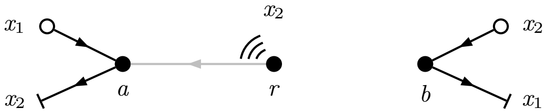
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



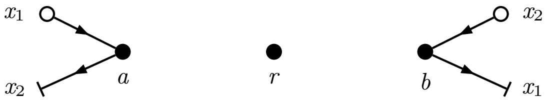
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



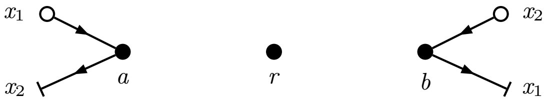
Routing

Exchanging 2 bits x_1 and x_2 with a routing approach:



Routing

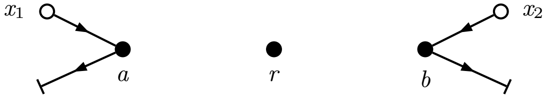
Exchanging 2 bits x_1 and x_2 with a routing approach:



sum rate: 2 bits in 4 transmissions

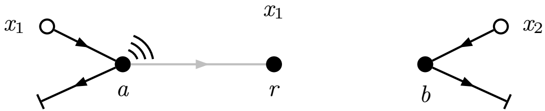
Network Coding

The relay node r can reach both a and b. Broadcasting saves one transmission.



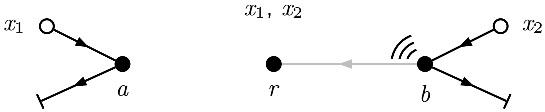
Network Coding

The relay node r can reach **both** a and b. Broadcasting saves one transmission.



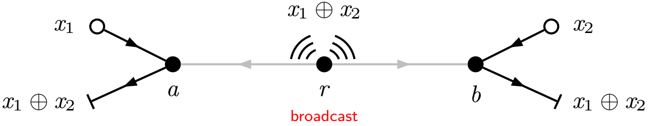
Network Coding

The relay node r can reach both a and b. Broadcasting saves one transmission.



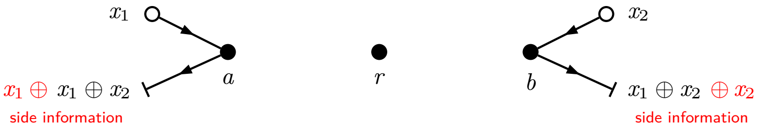
Network Coding

The relay node r can reach both a and b. Broadcasting saves one transmission.



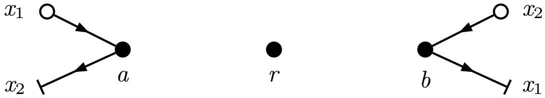
Network Coding

The relay node r can reach both a and b. Broadcasting saves one transmission.



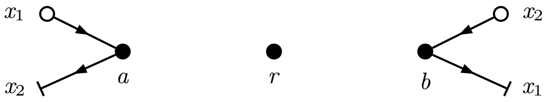
Network Coding

The relay node r can reach both a and b. Broadcasting saves one transmission.



Network Coding

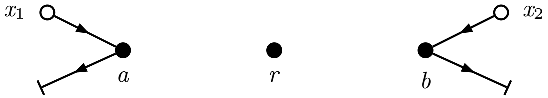
The relay node r can reach both a and b. Broadcasting saves one transmission.



sum rate: 2 bits in 3 transmissions

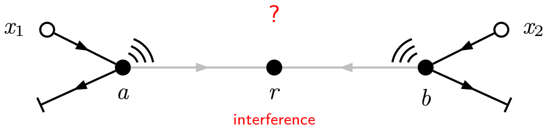
Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.



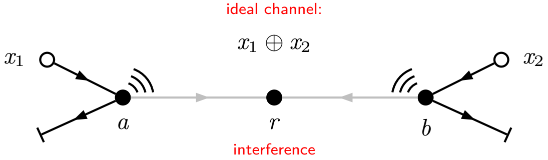
Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.



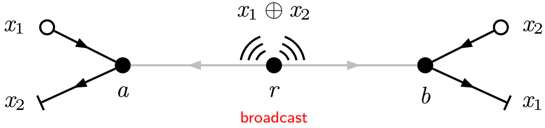
Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.



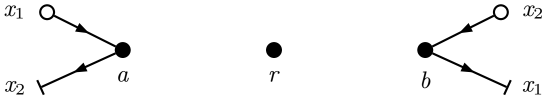
Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.



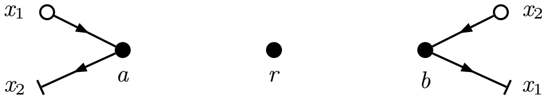
Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.



Physical-Layer Network Coding

The relay node doesn't need to know the individual bits.

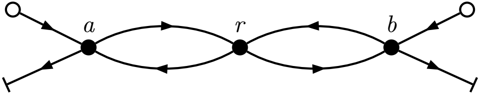


sum rate: 2 bits in 2 transmissions (for ideal channel)

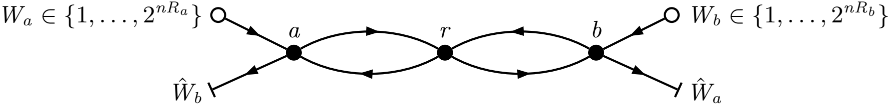
Obvious Question

How does it work for more realistic channel models?

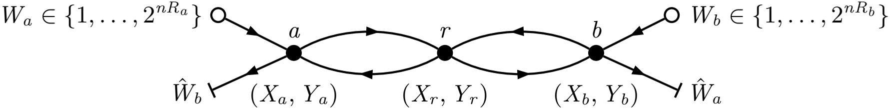
Gaussian Channel Model



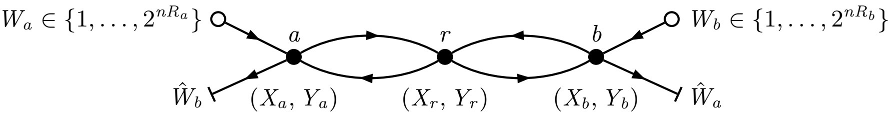
Gaussian Channel Model



Gaussian Channel Model



Gaussian Channel Model

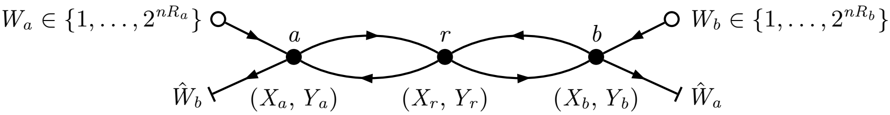


Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

Gaussian Channel Model



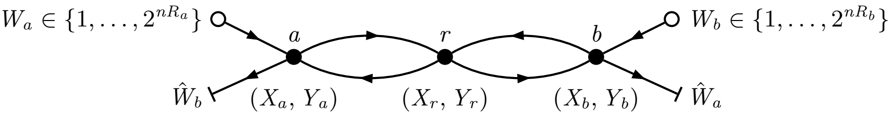
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes

Gaussian Channel Model



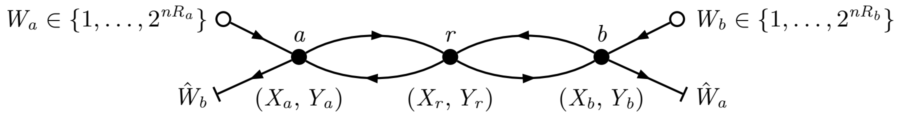
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \dots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b

Gaussian Channel Model



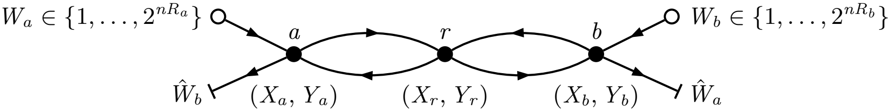
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b
- Noise: zero mean, variance σ_a^2, σ_r^2, σ_b^2

Gaussian Channel Model



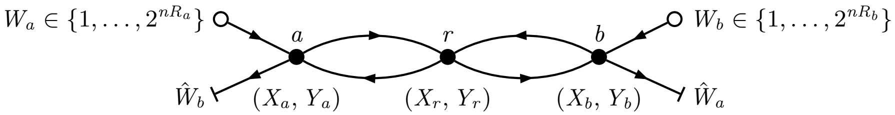
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b
- Noise: zero mean, variance σ_a^2, σ_r^2, σ_b^2
- Error probability:
  $p_e = \Pr(\{\hat{W}_a \neq W_a\} \cup \{\hat{W}_b \neq W_b\})$

Gaussian Channel Model



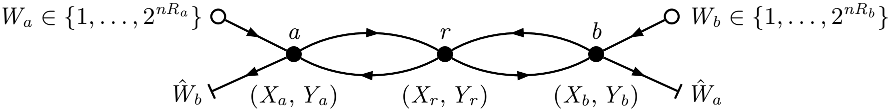
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b
- Noise: zero mean, variance σ_a^2, σ_r^2, σ_b^2
- Error probability:
  $p_e = \Pr(\{\hat{W}_a \neq W_a\} \cup \{\hat{W}_b \neq W_b\})$
- Uplink: a and b transmit to r.

Gaussian Channel Model



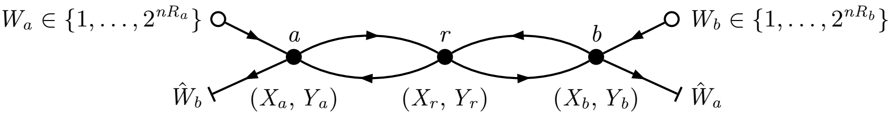
Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b
- Noise: zero mean, variance σ_a^2, σ_r^2, σ_b^2
- Error probability: $p_e = \Pr(\{\hat{W}_a \neq W_a\} \cup \{\hat{W}_b \neq W_b\})$
- Uplink: a and b transmit to r.
- Downlink: r broadcasts to a and b.

Gaussian Channel Model



Additive Gaussian noise:

$$Y_a = X_r + Z_a$$
$$Y_r = X_a + X_b + Z_r$$
$$Y_b = X_r + Z_b$$

Successive channel uses:
$\boldsymbol{X}_a = (X_{a,1}, \ldots, X_{a,n})$, etc.

- Full-duplex nodes
- Power constraint: P_a, P_r, P_b
- Noise: zero mean, variance σ_a^2, σ_r^2, σ_b^2
- Error probability: $p_e = \Pr(\{\hat{W}_a \neq W_a\} \cup \{\hat{W}_b \neq W_b\})$
- Uplink: a and b transmit to r.
- Downlink: r broadcasts to a and b.

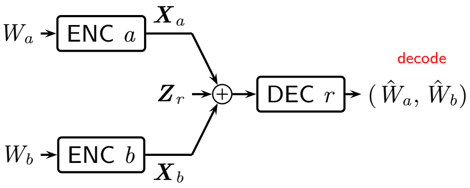
What (R_a, R_b) are achievable ($p_e \to 0$ as $n \to \infty$)?

Decode-and-forward (DF)

See [Rankov-Wittneben '05], [Knopp '06].

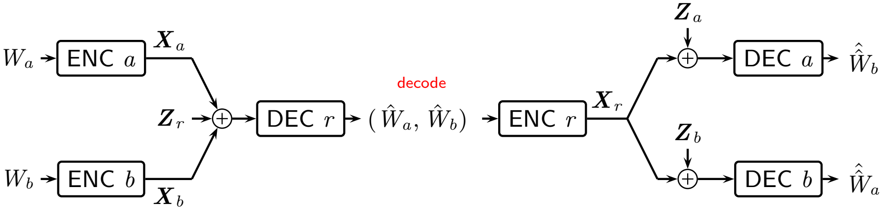
Decode-and-forward (DF)

See [Rankov-Wittneben '05], [Knopp '06].



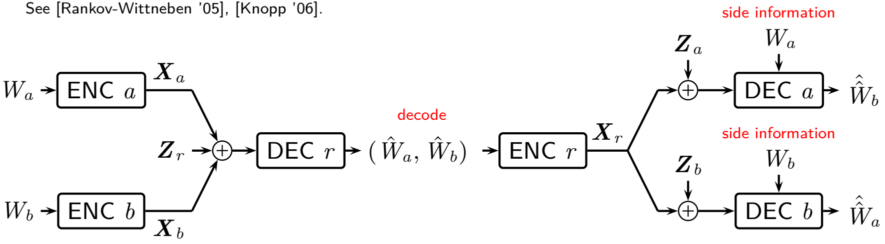
Decode-and-forward (DF)

See [Rankov-Wittneben '05], [Knopp '06].



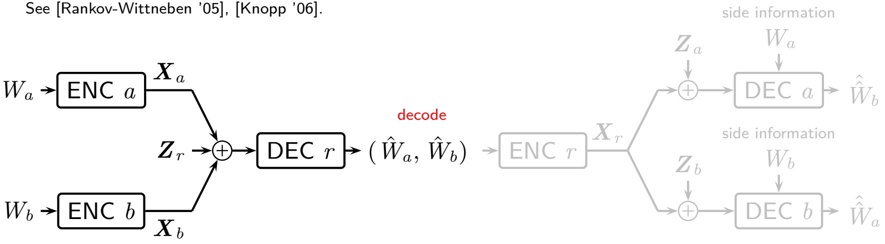
Decode-and-forward (DF)

See [Rankov-Wittneben '05], [Knopp '06].



Decode-and-forward (DF)

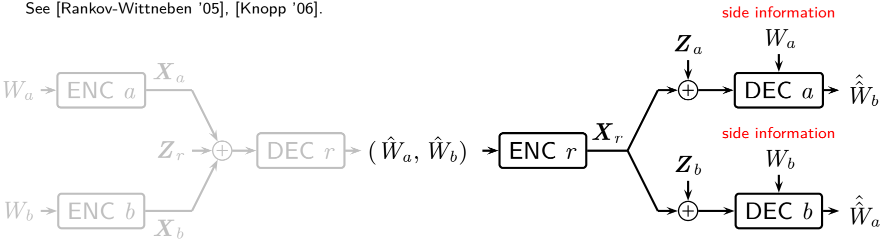
See [Rankov-Wittneben '05], [Knopp '06].



- Multi-acces channel

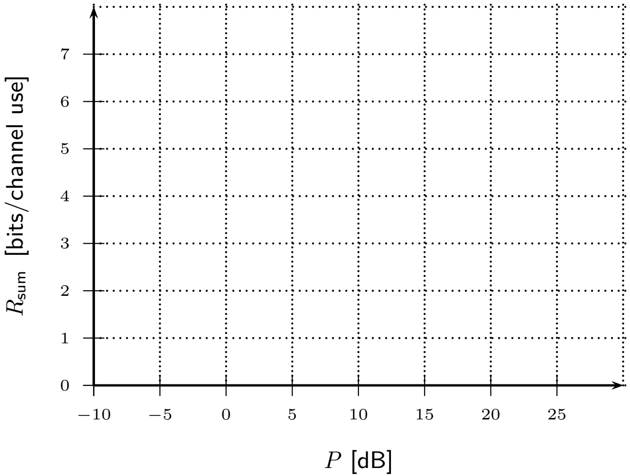
Decode-and-forward (DF)

See [Rankov-Wittneben '05], [Knopp '06].



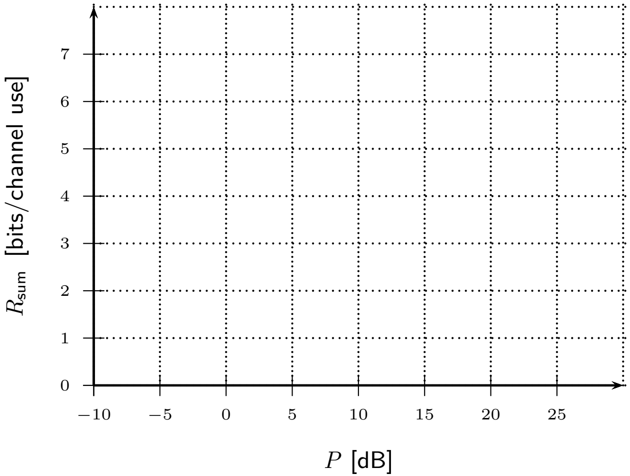
- Multi-acces channel
- Broadcasting with user side information

Sum Rate $R_{\text{SUM}} = R_a + R_b$



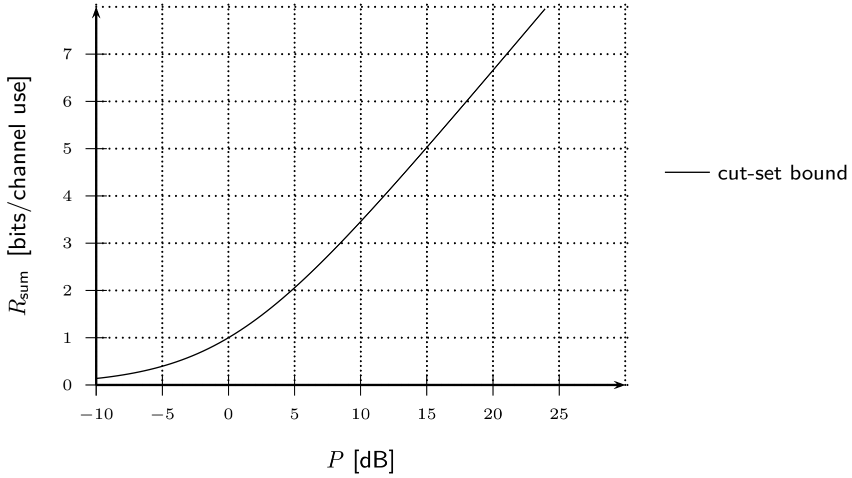
Sum Rate $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



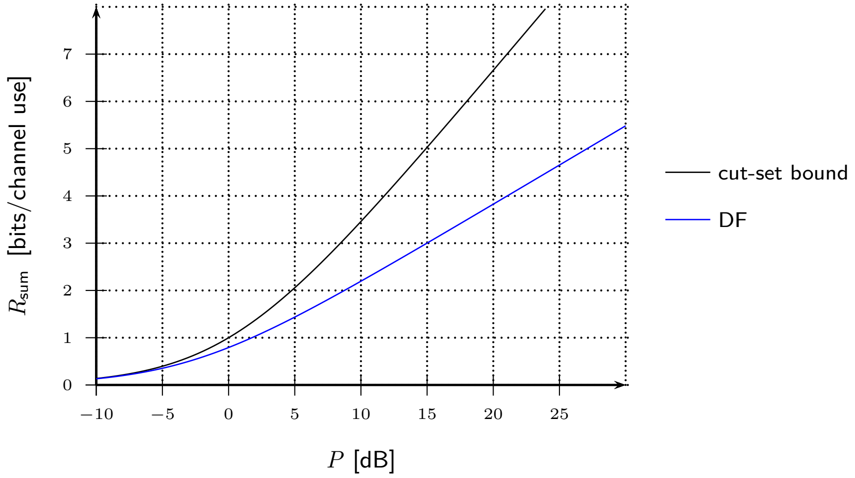
Sum Rate $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



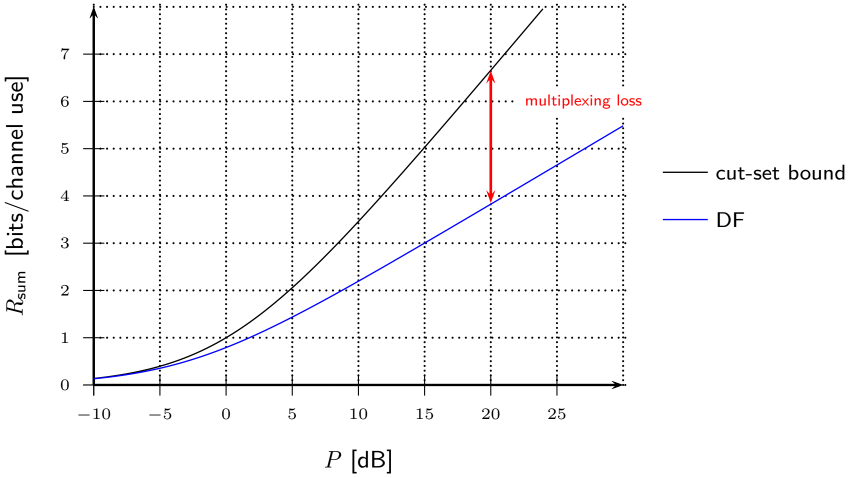
Sum Rate $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



SUM RATE $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



PNC with Structured Codes

PNC with Structured Codes

- Multiplexing loss: the relay tries to understand something that it doesn't really need to know, i.e., both messages individually.

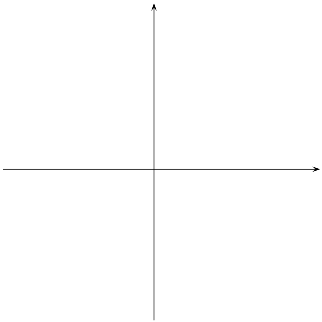
PNC with Structured Codes

- Multiplexing loss: the relay tries to understand something that it doesn't really need to know, i.e., both messages individually.
- PNC idea: the relay should only try to understand a combination of the transmitted messages.

PNC with Structured Codes

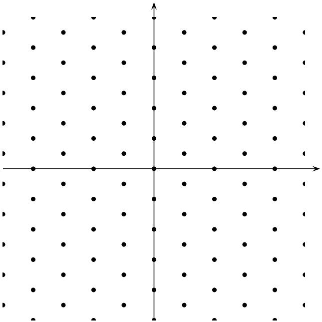
- Multiplexing loss: the relay tries to understand something that it doesn't really need to know, i.e., both messages individually.
- PNC idea: the relay should only try to understand a combination of the transmitted messages.
- Use structured codes: Voronoi codes

Lattices and Voronoi Codes



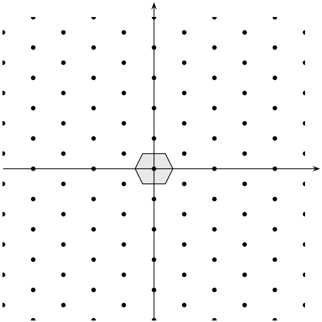
Lattices and Voronoi Codes

- (coding) Lattice Λ_c



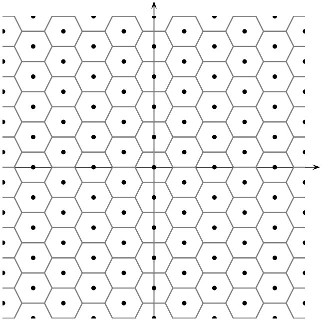
Lattices and Voronoi Codes

- (coding) Lattice Λ_c
- Fundamental (Voronoi) region $\mathcal{R}_V(\Lambda_c)$



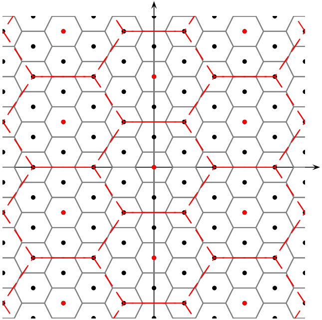
Lattices and Voronoi Codes

- (coding) Lattice Λ_c
- Fundamental (Voronoi) region $\mathcal{R}_V(\Lambda_c)$
- Voronoi regions



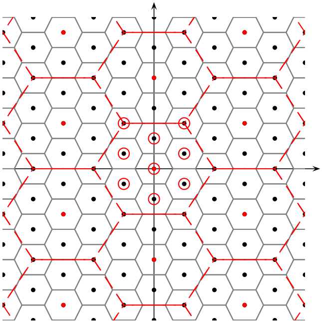
Lattices and Voronoi Codes

- (coding) Lattice Λ_c
- Fundamental (Voronoi) region $\mathcal{R}_V(\Lambda_c)$
- Voronoi regions
- Nested (shaping) lattice $\Lambda \subset \Lambda_c$

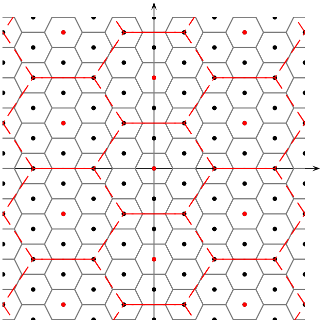


Lattices and Voronoi Codes

- (coding) Lattice Λ_c
- Fundamental (Voronoi) region $\mathcal{R}_V(\Lambda_c)$
- Voronoi regions
- Nested (shaping) lattice $\Lambda \subset \Lambda_c$
- Voronoi code:
  $\mathcal{C}(\Lambda_c/\Lambda) = (\Lambda_c + \boldsymbol{t}) \cap \mathcal{R}_V(\Lambda)$

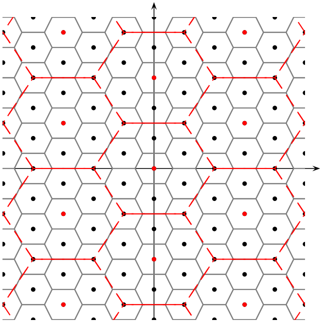


The Basic Idea



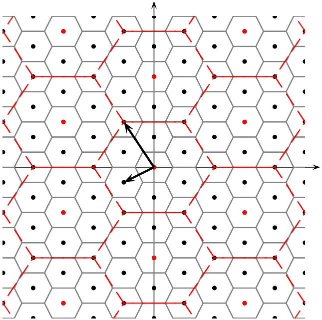
The Basic Idea

- Users a and b use the same Voronoi code $\mathcal{C}(\Lambda_c/\Lambda)$.



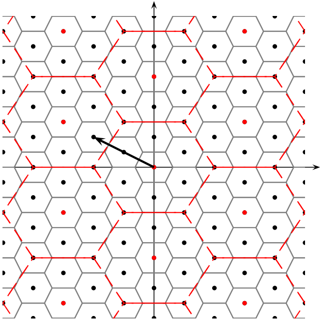
The Basic Idea

- Users a and b use the same Voronoi code $\mathcal{C}(\Lambda_c/\Lambda)$.
- Uplink: users transmit $\boldsymbol{V}_a, \boldsymbol{V}_b \in \mathcal{C}(\Lambda_c/\Lambda)$.



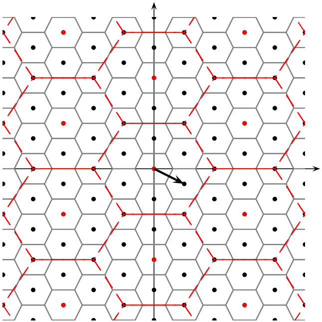
The Basic Idea

- Users a and b use the same Voronoi code $\mathcal{C}(\Lambda_c/\Lambda)$.
- Uplink: users transmit $\boldsymbol{V}_a, \boldsymbol{V}_b \in \mathcal{C}(\Lambda_c/\Lambda)$.
- Relay receives $\boldsymbol{V}_a + \boldsymbol{V}_b \in \Lambda_c$.



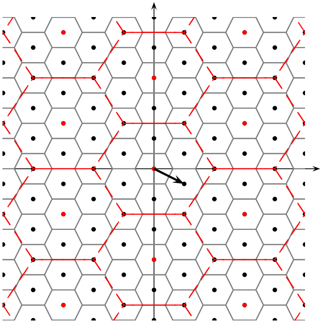
The Basic Idea

- Users a and b use the same Voronoi code $\mathcal{C}(\Lambda_c/\Lambda)$.
- Uplink: users transmit $\boldsymbol{V}_a, \boldsymbol{V}_b \in \mathcal{C}(\Lambda_c/\Lambda)$.
- Relay receives $\boldsymbol{V}_a + \boldsymbol{V}_b \in \Lambda_c$.
- Relay computes $\boldsymbol{V} = (\boldsymbol{V}_a + \boldsymbol{V}_b) \mod \Lambda \in \mathcal{C}(\Lambda_c/\Lambda)$.



The Basic Idea

- Users a and b use the same Voronoi code $\mathcal{C}(\Lambda_c/\Lambda)$.
- Uplink: users transmit $\boldsymbol{V}_a, \boldsymbol{V}_b \in \mathcal{C}(\Lambda_c/\Lambda)$.
- Relay receives $\boldsymbol{V}_a + \boldsymbol{V}_b \in \Lambda_c$.
- Relay computes $\boldsymbol{V} = (\boldsymbol{V}_a + \boldsymbol{V}_b) \mod \Lambda \in \mathcal{C}(\Lambda_c/\Lambda)$.
- Side information allows each user to recover the lattice point of the other user based on $\boldsymbol{V}$.



Result for Arbitrary Channel Conditions

Result for Arbitrary Channel Conditions

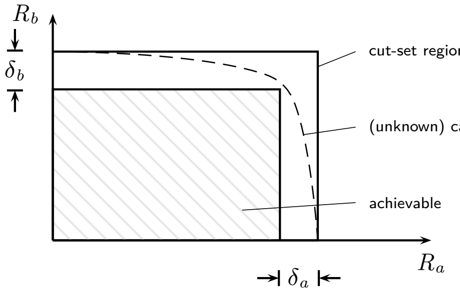
Theorem

The capacity region is achievable to within $1/2$ *bits per dimension.*

Result for Arbitrary Channel Conditions

Theorem

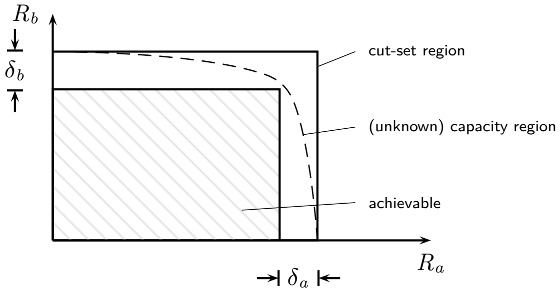
The capacity region is achievable to within $1/2$ bits per dimension.



Result for Arbitrary Channel Conditions

Theorem

The capacity region is achievable to within $1/2$ bits per dimension.

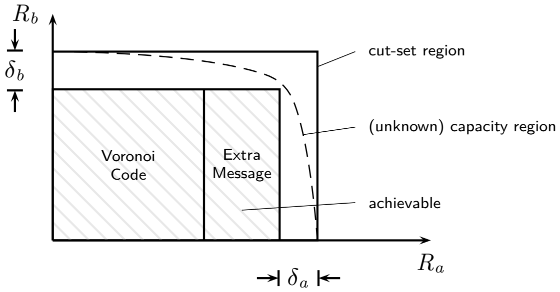


Achievability strategy:

Result for Arbitrary Channel Conditions

Theorem

The capacity region is achievable to within $1/2$ bits per dimension.



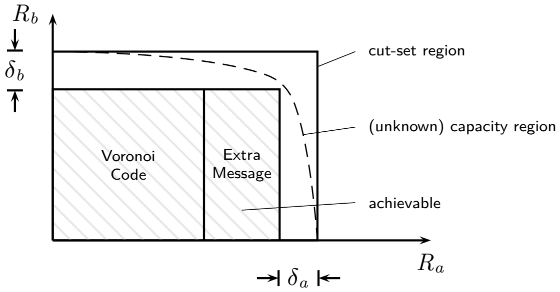
Achievability strategy:

1. Uplink: rate splitting, Voronoi code, and superposition coding

Result for Arbitrary Channel Conditions

Theorem

The capacity region is achievable to within $1/2$ *bits per dimension.*



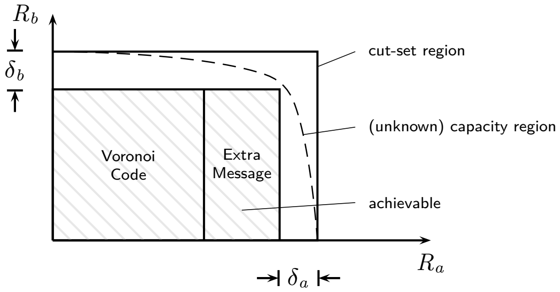
Achievability strategy:

1. Uplink: rate splitting, Voronoi code, and superposition coding
2. Downlink: two-dimensional random codebook

Result for Arbitrary Channel Conditions

Theorem

The capacity region is achievable to within $1/2$ bits per dimension.

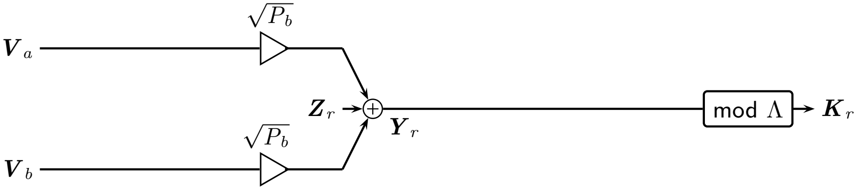


Achievability strategy:

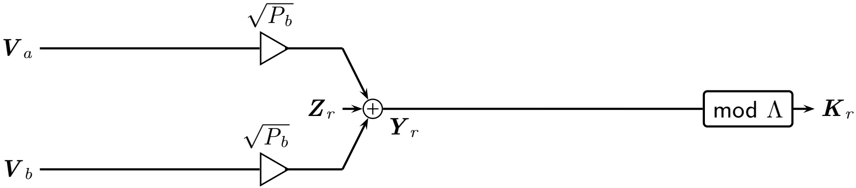
1. Uplink: rate splitting, Voronoi code, and superposition coding
2. Downlink: two-dimensional random codebook

(see also [Nam-Chung-Lee '10])

Uplink

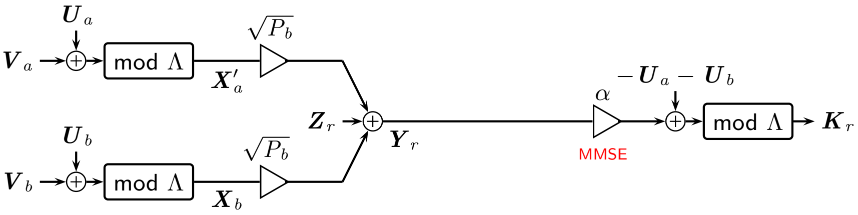

$\sqrt{P_b}$
$\boldsymbol{V}_a$
$\boldsymbol{Z}_r$
$\boldsymbol{Y}_r$
mod Λ
$\boldsymbol{K}_r$
$\sqrt{P_b}$
$\boldsymbol{V}_b$

Uplink



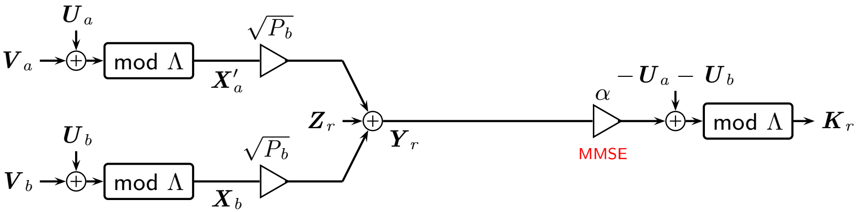
- Lattice decoding: $Q_{\Lambda_c}(\boldsymbol{K}_r)$

Uplink



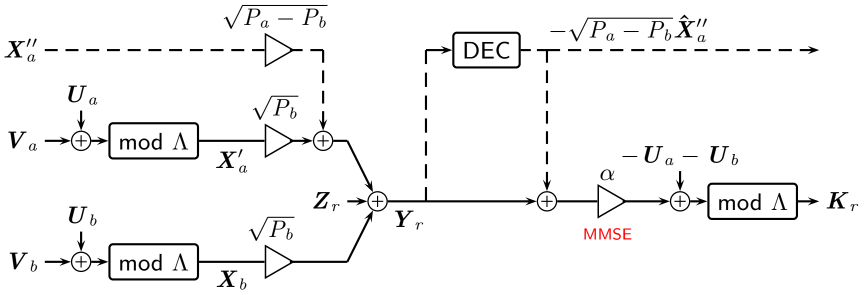
- Lattice decoding: $Q_{\Lambda_c}(\boldsymbol{K}_r)$
- Dithering and linear MMSE estimation for effective noise reduction

Uplink



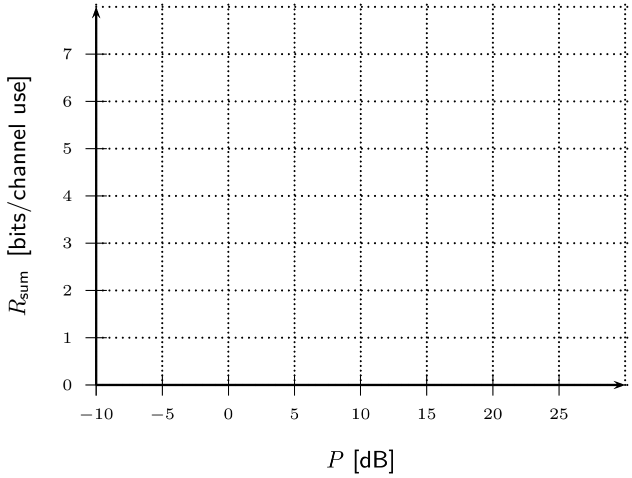
- Lattice decoding: $Q_{\Lambda_c}(\boldsymbol{K}_r)$
- Dithering and linear MMSE estimation for effective noise reduction
- $\boldsymbol{K}_r = (\alpha \boldsymbol{Y}_r - \boldsymbol{U}_a - \boldsymbol{U}_b) \bmod \Lambda = (\boldsymbol{V}_a + \boldsymbol{V}_b + \tilde{\boldsymbol{Z}}_r) \bmod \Lambda$

Uplink



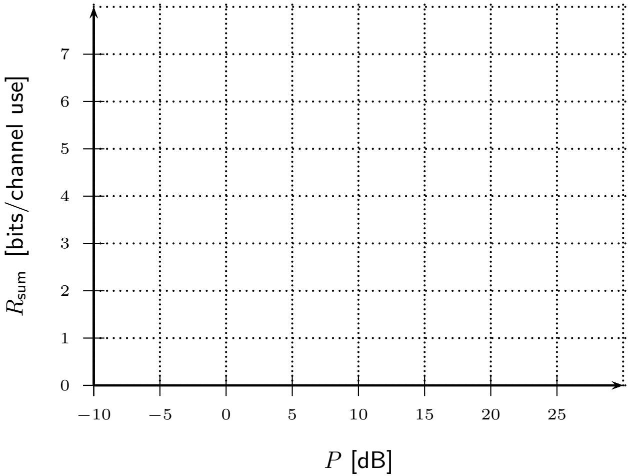
- Lattice decoding: $Q_{\Lambda_c}(\boldsymbol{K}_r)$
- Dithering and linear MMSE estimation for effective noise reduction
- $\boldsymbol{K}_r = (\alpha \boldsymbol{Y}_r - \boldsymbol{U}_a - \boldsymbol{U}_b) \mod \Lambda = (\boldsymbol{V}_a + \boldsymbol{V}_b + \tilde{\boldsymbol{Z}}_r) \mod \Lambda$
- Superposition coding with successive cancellation

SUM RATE $R_{\text{SUM}} = R_a + R_b$



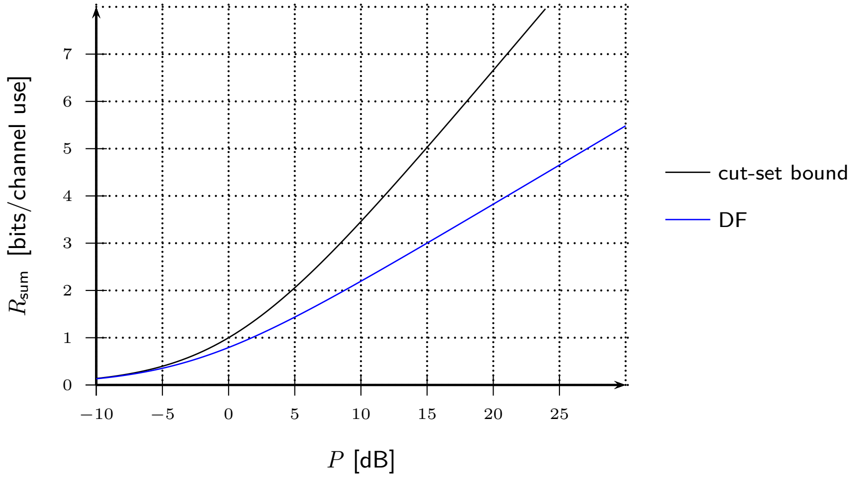
Sum Rate $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



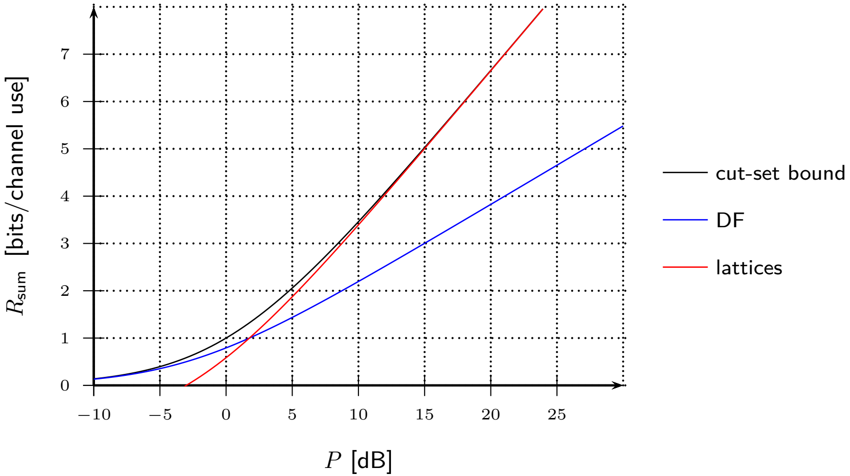
Sum Rate $R_{\text{SUM}} = R_a + R_b$

Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$

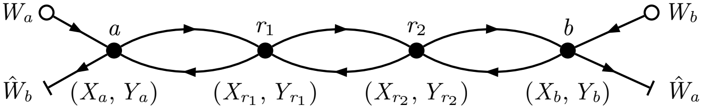


Sum Rate $R_{\text{SUM}} = R_a + R_b$

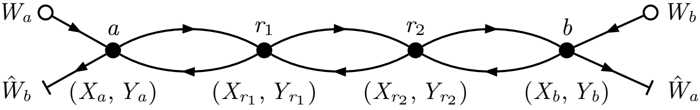
Symmetric case: $P_a = P_r = P_b = P$ and $\sigma_a^2 = \sigma_r^2 = \sigma_b^2 = 1$



The Separated Two-Way Two-Relay Channel

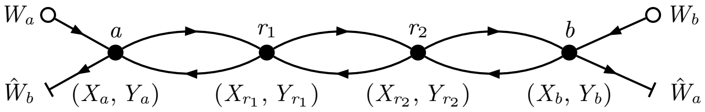


The Separated Two-Way Two-Relay Channel



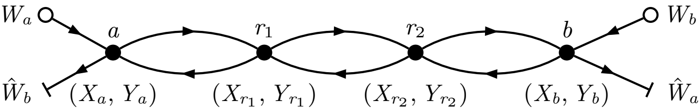
- Multi-hop extension of the separated two-way relay channel

The Separated Two-Way Two-Relay Channel



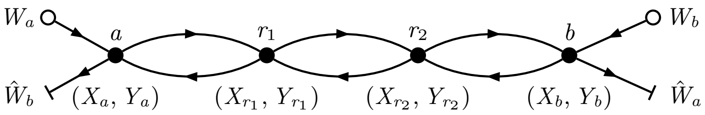
- Multi-hop extension of the separated two-way relay channel
- Both relays are necessary to enable the message exchange.

The Separated Two-Way Two-Relay Channel



- Multi-hop extension of the separated two-way relay channel
- Both relays are necessary to enable the message exchange.
- No sinks or sources at the relays

The Separated Two-Way Two-Relay Channel



- Multi-hop extension of the separated two-way relay channel
- Both relays are necessary to enable the message exchange.
- No sinks or sources at the relays
- Transmission strategy: physical-layer network coding with nested Voronoi codes

Another Toy Problem

Another Toy Problem

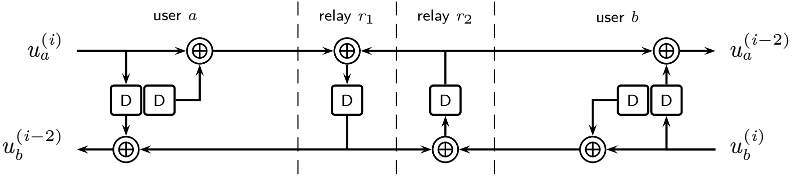
- No noise, binary inputs and outputs, finite field physical layer, full-duplex nodes

Another Toy Problem

- No noise, binary inputs and outputs, finite field physical layer, full-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$

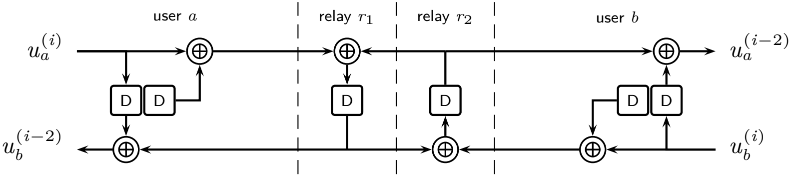
Another Toy Problem

- No noise, binary inputs and outputs, finite field physical layer, full-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



Another Toy Problem

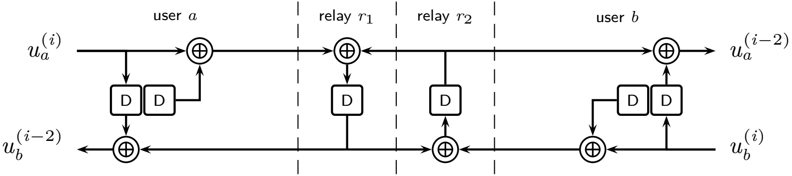
- No noise, binary inputs and outputs, finite field physical layer, full-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



- Each user “sees” the packet stream of the other user – delayed by two blocks.

Another Toy Problem

- No noise, binary inputs and outputs, finite field physical layer, full-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



- Each user “sees” the packet stream of the other user – delayed by two blocks.
- User rates approach 1 bit per channel use for many blocks.

The Gaussian Case

The Gaussian Case

Channel model:

$$\begin{aligned} Y_a &= X_{r_1} + Z_a \\ Y_{r_1} &= X_a + X_{r_2} + Z_{r_1} \\ Y_{r_2} &= X_{r_1} + X_b + Z_{r_2} \\ Y_b &= X_{r_2} + Z_b \end{aligned}$$

The Gaussian Case

Channel model:

$$
\begin{aligned}
Y_a &= X_{r_1} + Z_a \\
Y_{r_1} &= X_a + X_{r_2} + Z_{r_1} \\
Y_{r_2} &= X_{r_1} + X_b + Z_{r_2} \\
Y_b &= X_{r_2} + Z_b
\end{aligned}
$$

- Same power constraint P for all nodes

The Gaussian Case

Channel model:

$$Y_a = X_{r_1} + Z_a$$
$$Y_{r_1} = X_a + X_{r_2} + Z_{r_1}$$
$$Y_{r_2} = X_{r_1} + X_b + Z_{r_2}$$
$$Y_b = X_{r_2} + Z_b$$

- Same power constraint P for all nodes
- Gaussian noise with zero mean, variances σ_a^2, $\sigma_{r_1}^2$, $\sigma_{r_2}^2$, σ_b^2

The Gaussian Case

Channel model:

$$
\begin{aligned}
Y_a &= X_{r_1} + Z_a \\
Y_{r_1} &= X_a + X_{r_2} + Z_{r_1} \\
Y_{r_2} &= X_{r_1} + X_b + Z_{r_2} \\
Y_b &= X_{r_2} + Z_b
\end{aligned}
$$

- Same power constraint P for all nodes
- Gaussian noise with zero mean, variances σ_a^2, $\sigma_{r_1}^2$, $\sigma_{r_2}^2$, σ_b^2
- Successive channel uses, transmission block i: $\boldsymbol{X}_a^{(i)}$, etc.

The Gaussian Case

Channel model:

$$Y_a = X_{r_1} + Z_a$$
$$Y_{r_1} = X_a + X_{r_2} + Z_{r_1}$$
$$Y_{r_2} = X_{r_1} + X_b + Z_{r_2}$$
$$Y_b = X_{r_2} + Z_b$$

- Same power constraint P for all nodes
- Gaussian noise with zero mean, variances σ_a^2, $\sigma_{r_1}^2$, $\sigma_{r_2}^2$, σ_b^2
- Successive channel uses, transmission block i: $\boldsymbol{X}_a^{(i)}$, etc.

Theorem

One can achieve all rate pairs (R_a, R_b) *satisfying*

$$R_a < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \mathrm{C}\left(P/\sigma_b^2\right)\right)$$
$$R_b < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \mathrm{C}\left(P/\sigma_a^2\right)\right)$$

where $\tilde{\mathrm{C}}(x) \triangleq \log_2(1/2 + x)/2$ *and* $\mathrm{C}(x) \triangleq \log_2(1 + x)/2$.

The Gaussian Case

Channel model:

$$Y_a = X_{r_1} + Z_a$$
$$Y_{r_1} = X_a + X_{r_2} + Z_{r_1}$$
$$Y_{r_2} = X_{r_1} + X_b + Z_{r_2}$$
$$Y_b = X_{r_2} + Z_b$$

- Same power constraint P for all nodes
- Gaussian noise with zero mean, variances σ_a^2, $\sigma_{r_1}^2$, $\sigma_{r_2}^2$, σ_b^2
- Successive channel uses, transmission block i: $\boldsymbol{X}_a^{(i)}$, etc.

Theorem

One can achieve all rate pairs (R_a, R_b) *satisfying*

$$R_a < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \mathrm{C}\left(P/\sigma_b^2\right)\right)$$
$$R_b < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \mathrm{C}\left(P/\sigma_a^2\right)\right)$$

where $\tilde{\mathrm{C}}(x) \triangleq \log_2(1/2 + x)/2$ *and* $\mathrm{C}(x) \triangleq \log_2(1+x)/2$.

- Capacity region to within 1/2 bit per dimension

The Gaussian Case

Channel model:

$$Y_a = X_{r_1} + Z_a$$
$$Y_{r_1} = X_a + X_{r_2} + Z_{r_1}$$
$$Y_{r_2} = X_{r_1} + X_b + Z_{r_2}$$
$$Y_b = X_{r_2} + Z_b$$

- Same power constraint P for all nodes
- Gaussian noise with zero mean, variances σ_a^2, $\sigma_{r_1}^2$, $\sigma_{r_2}^2$, σ_b^2
- Successive channel uses, transmission block i: $\boldsymbol{X}_a^{(i)}$, etc.

Theorem

One can achieve all rate pairs (R_a, R_b) *satisfying*

$$R_a < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \mathrm{C}\left(P/\sigma_b^2\right)\right)$$
$$R_b < \min\left(\tilde{\mathrm{C}}\left(P/\sigma_{r_2}^2\right), \tilde{\mathrm{C}}\left(P/\sigma_{r_1}^2\right), \mathrm{C}\left(P/\sigma_a^2\right)\right)$$

where $\tilde{\mathrm{C}}(x) \triangleq \log_2(1/2 + x)/2$ *and* $\mathrm{C}(x) \triangleq \log_2(1 + x)/2$.

- Capacity region to within 1/2 bit per dimension
- Essentially optimal at high SNR

MLAN Conversion

See [Erez-Zamir '08].

MLAN Conversion

See [Erez-Zamir '08].

Blockwise modulo-lattice additive noise channel:

$$\begin{aligned}
\tilde{\boldsymbol{Y}}_a^{(i)} &= (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{Z}}_a^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_{r_1}^{(i)} &= (\tilde{\boldsymbol{X}}_a^{(i)} + \tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_{r_1}^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_{r_2}^{(i)} &= (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{X}}_b^{(i)} + \tilde{\boldsymbol{Z}}_{r_2}^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_b^{(i)} &= (\tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_b^{(i)}) \mod \Lambda
\end{aligned}$$

where

MLAN Conversion

See [Erez-Zamir '08].

Blockwise modulo-lattice additive noise channel:

$$\tilde{\boldsymbol{Y}}_a^{(i)} = (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{Z}}_a^{(i)}) \bmod \Lambda$$
$$\tilde{\boldsymbol{Y}}_{r_1}^{(i)} = (\tilde{\boldsymbol{X}}_a^{(i)} + \tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_{r_1}^{(i)}) \bmod \Lambda$$
$$\tilde{\boldsymbol{Y}}_{r_2}^{(i)} = (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{X}}_b^{(i)} + \tilde{\boldsymbol{Z}}_{r_2}^{(i)}) \bmod \Lambda$$
$$\tilde{\boldsymbol{Y}}_b^{(i)} = (\tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_b^{(i)}) \bmod \Lambda$$

where

- $\tilde{\boldsymbol{X}}_a^{(i)}, \tilde{\boldsymbol{X}}_{r_1}^{(i)}, \tilde{\boldsymbol{X}}_{r_2}^{(i)}, \tilde{\boldsymbol{X}}_b^{(i)} \in \mathcal{R}_V(\Lambda)$ are the input signals in block i.

MLAN Conversion

See [Erez-Zamir '08].

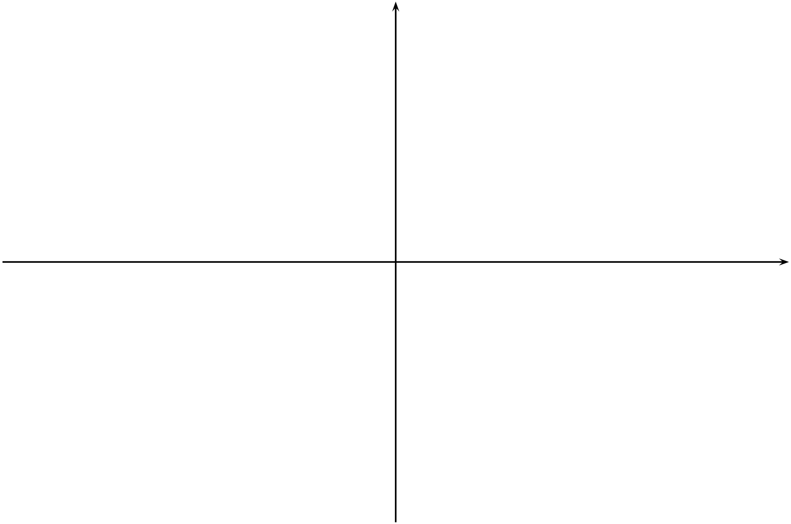
Blockwise modulo-lattice additive noise channel:

$$\begin{aligned}
\tilde{\boldsymbol{Y}}_a^{(i)} &= (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{Z}}_a^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_{r_1}^{(i)} &= (\tilde{\boldsymbol{X}}_a^{(i)} + \tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_{r_1}^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_{r_2}^{(i)} &= (\tilde{\boldsymbol{X}}_{r_1}^{(i)} + \tilde{\boldsymbol{X}}_b^{(i)} + \tilde{\boldsymbol{Z}}_{r_2}^{(i)}) \mod \Lambda \\
\tilde{\boldsymbol{Y}}_b^{(i)} &= (\tilde{\boldsymbol{X}}_{r_2}^{(i)} + \tilde{\boldsymbol{Z}}_b^{(i)}) \mod \Lambda
\end{aligned}$$

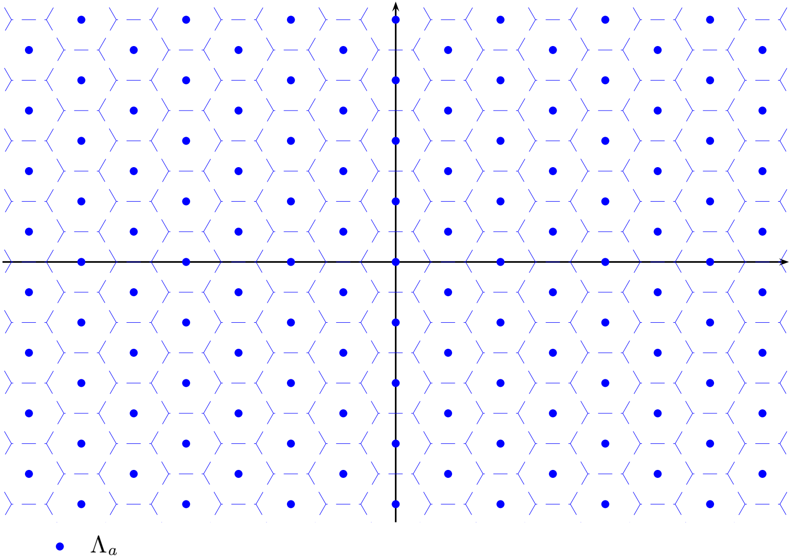
where

- $\tilde{\boldsymbol{X}}_a^{(i)}, \tilde{\boldsymbol{X}}_{r_1}^{(i)}, \tilde{\boldsymbol{X}}_{r_2}^{(i)}, \tilde{\boldsymbol{X}}_b^{(i)} \in \mathcal{R}_V(\Lambda)$ are the input signals in block i.
- Effective noise $\tilde{\boldsymbol{Z}}_a^{(i)}, \tilde{\boldsymbol{Z}}_{r_1}^{(i)}, \tilde{\boldsymbol{Z}}_{r_2}^{(i)}, \tilde{\boldsymbol{Z}}_b^{(i)}$ is statistically independent of the inputs and has reduced effective noise power.

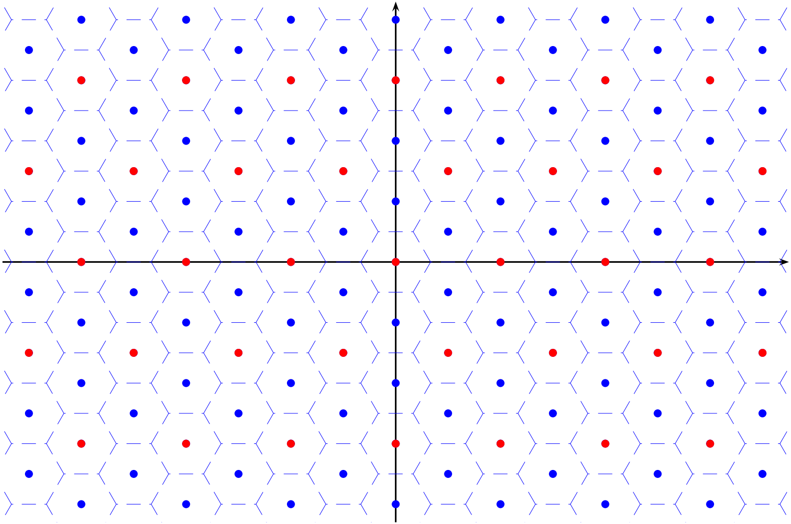
Nested Voronoi Codes



Nested Voronoi Codes

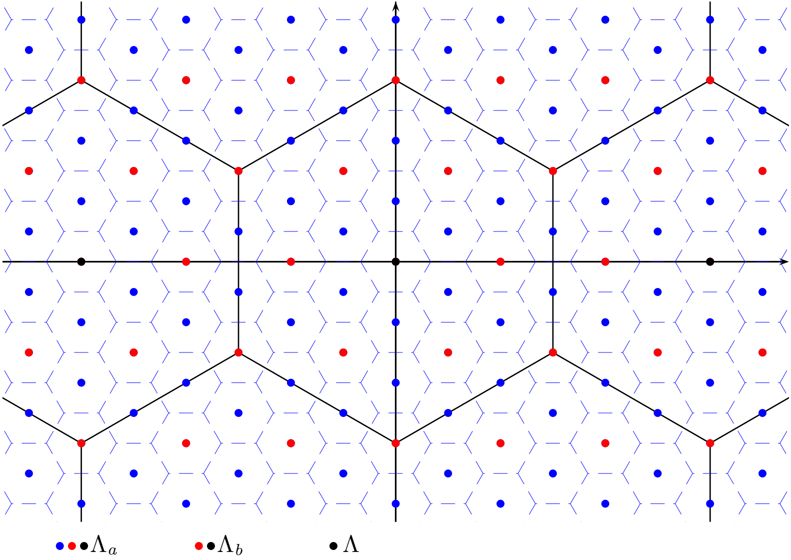


Nested Voronoi Codes

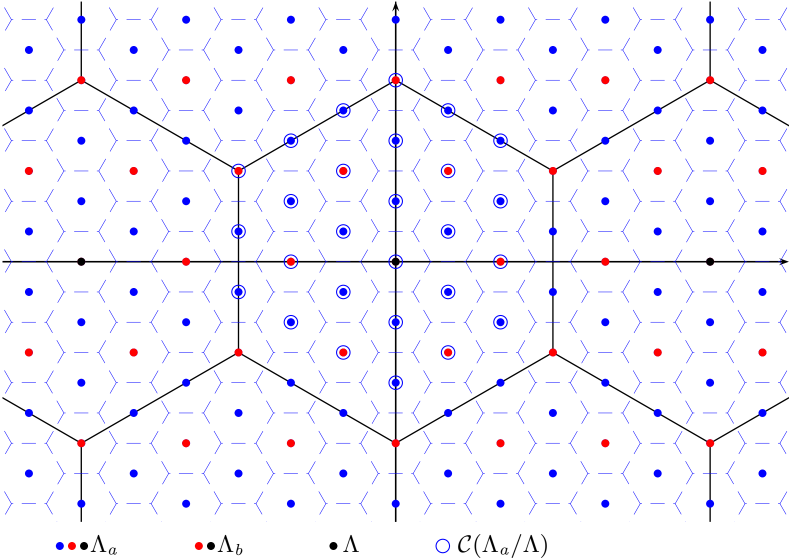


●● Λ_a ● Λ_b

Nested Voronoi Codes

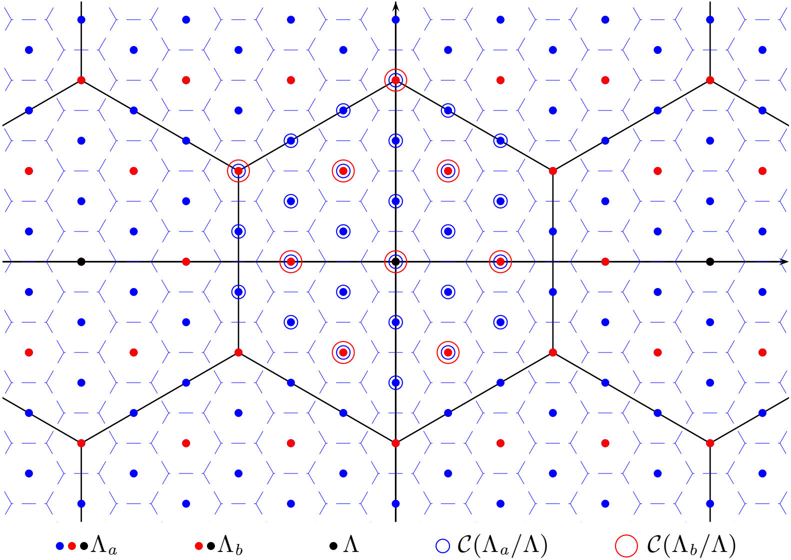


Nested Voronoi Codes

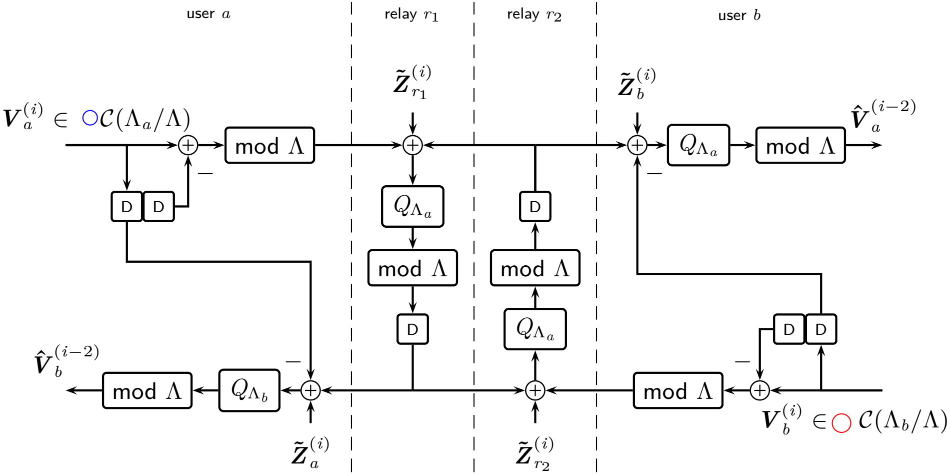


CHALMERS

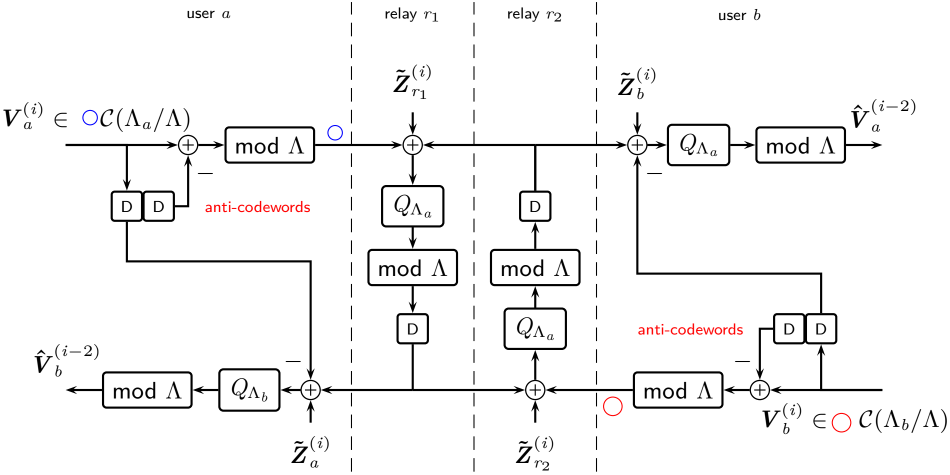
Nested Voronoi Codes



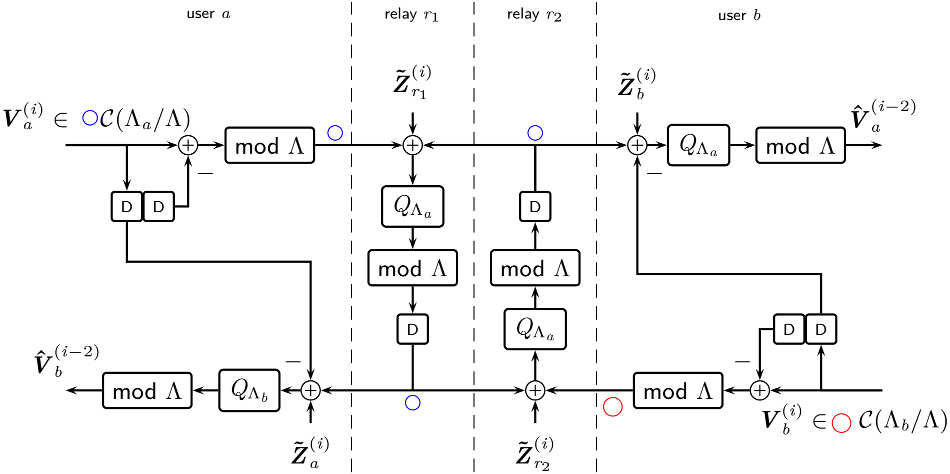
Block Diagram



Block Diagram

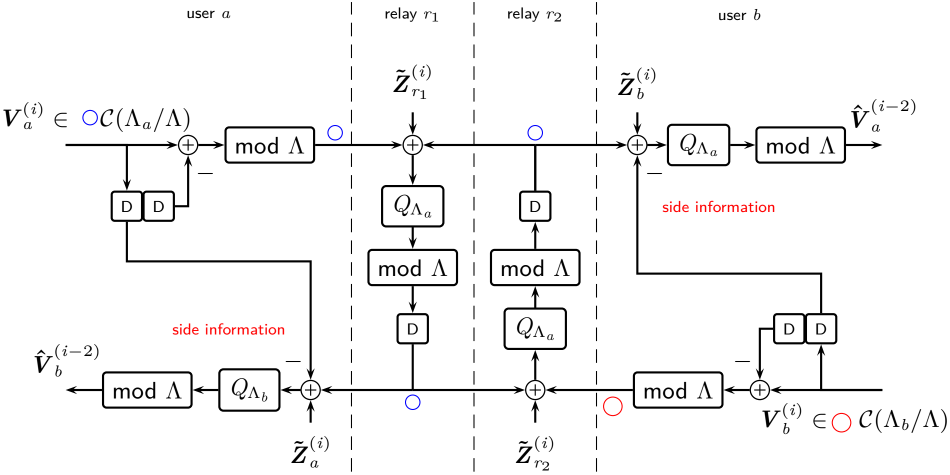


Block Diagram

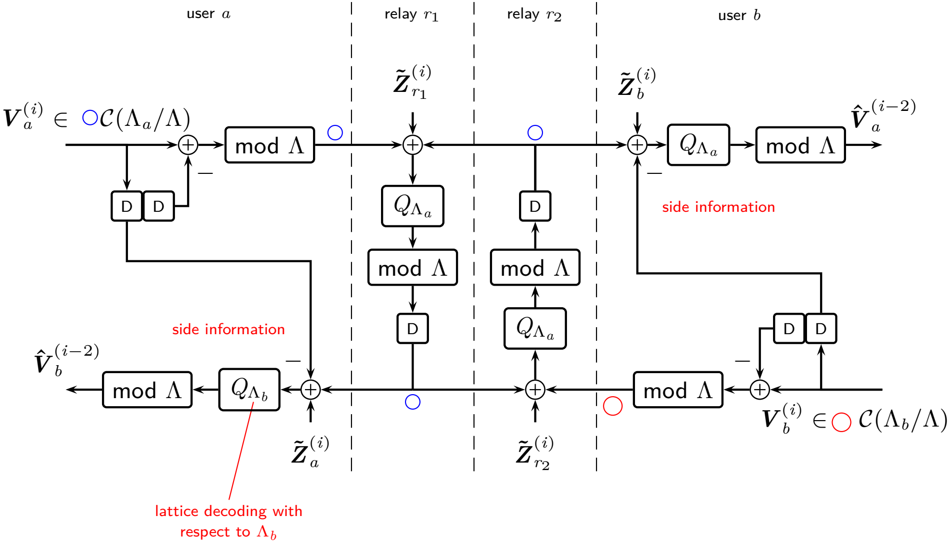


relays "protect" linear combinations of codewords

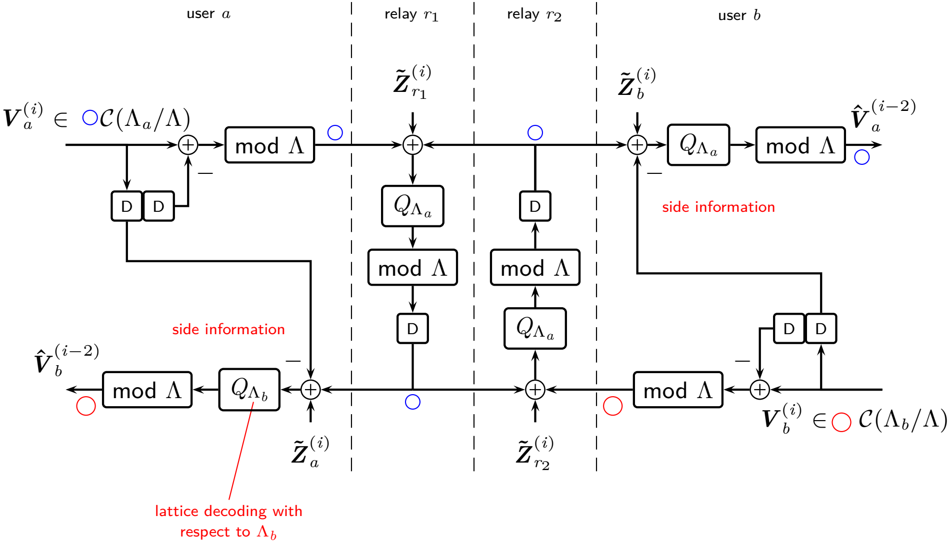
Block Diagram



Block Diagram



Block Diagram



Final Remarks

Final Remarks

- Superposition coding for unequal power constraints does not lead to a gap with respect to the upper bound in general.

Final Remarks

- Superposition coding for unequal power constraints does not lead to a gap with respect to the upper bound in general.
- Different shaping lattices might provide a solution.

Final Remarks

- Superposition coding for unequal power constraints does not lead to a gap with respect to the upper bound in general.
- Different shaping lattices might provide a solution.
- Generalization to L relays (assuming full separation) is possible.

Conclusion

Conclusion

- Structured codes appear to be a powerful tool to show achievability of rates in communication networks.

Conclusion

- Structured codes appear to be a powerful tool to show achievability of rates in communication networks.
- Involved principles: physical-layer network coding, broadcasting, using side information

Conclusion

- Structured codes appear to be a powerful tool to show achievability of rates in communication networks.
- Involved principles: physical-layer network coding, broadcasting, using side information
- In certain networks the interference of users can be harnessed.

Conclusion

- Structured codes appear to be a powerful tool to show achievability of rates in communication networks.
- Involved principles: physical-layer network coding, broadcasting, using side information
- In certain networks the interference of users can be harnessed.
- However, a strategy to show full achievability of the upper bound is still not available.

References

B. Rankov and A. Wittneben.
Spectral efficient signaling for half-duplex relay channels.
2005, Proc. of Asilomar Conference on Signals, Systems and Computers

R. Knopp.
Two-way radio networks with a star topology.
2006, International Zurich Seminar on Communications

U. Erez and R. Zamir.
A Modulo-Lattice Transformation for Multiple-Access Channels
2008, IEEE Convention of Electrical and Electronics Engineers in Israel

W. Nam and S.-Y. Chung and Y. Lee.
Capacity of the Gaussian Two-Way Relay Channel to within $1/2$ bit
2010, IEEE Transactions on Information Theory

Downlink

	1	$\cdots$	$2^{n(R_a - R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

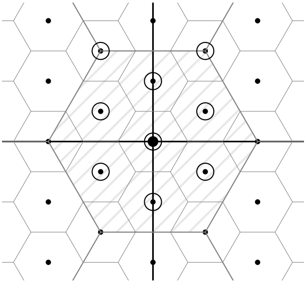
Downlink

	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$

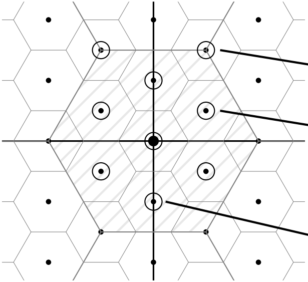


	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$

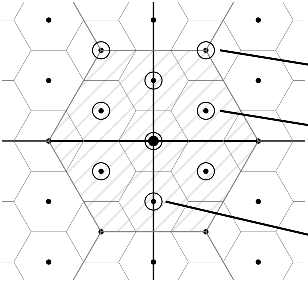


	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$



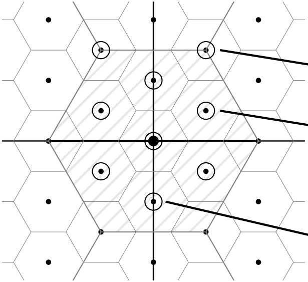
extra message

	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$



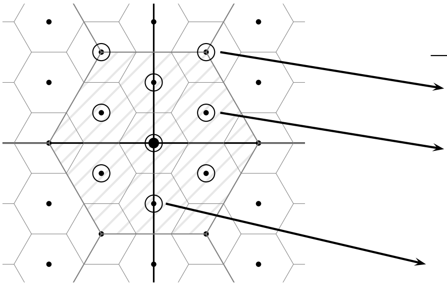
extra message

	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns
- User b decodes with respect to the whole codebook.

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$



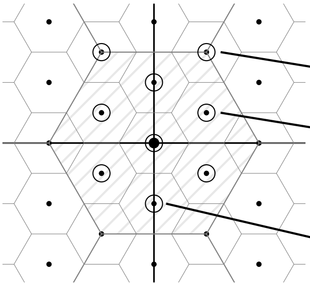
extra message

	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns
- User b decodes with respect to the whole codebook.
- User a decodes with respect to a column of the codebook.

Downlink

$\boldsymbol{V} \in \mathcal{C}(\Lambda_c/\Lambda)$



extra message

	1	$\cdots$	$2^{n(R_a-R_b)}$
1	$\boldsymbol{X}_r(1,1)$	$\cdots$	$\boldsymbol{X}_r(1,M)$
2	$\boldsymbol{X}_r(2,1)$	$\cdots$	$\boldsymbol{X}_r(2,M)$
$\vdots$	$\vdots$		$\vdots$
2^{nR_b}	$\boldsymbol{X}_r(N,1)$	$\cdots$	$\boldsymbol{X}_r(N,M)$

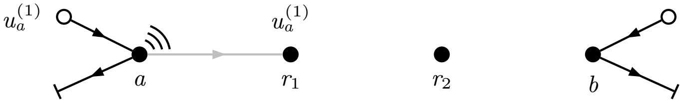
- Random codebook: $N = 2^{nR_b}$ rows, $M = 2^{n(R_a-R_b)}$ columns
- User b decodes with respect to the whole codebook.
- User a decodes with respect to a column of the codebook.
- Optimal downlink strategy.

Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$

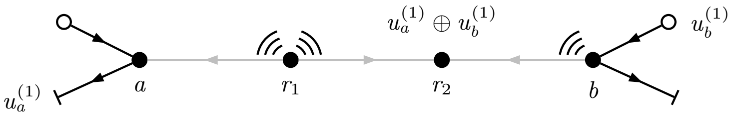
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



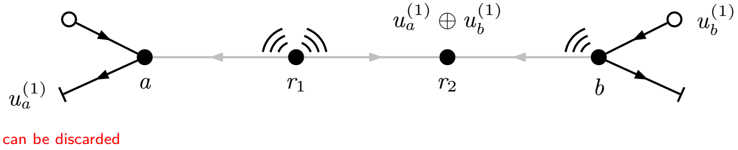
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \dots$ and $u_b^{(1)}, u_b^{(2)}, \dots$



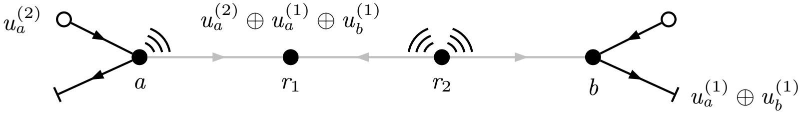
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



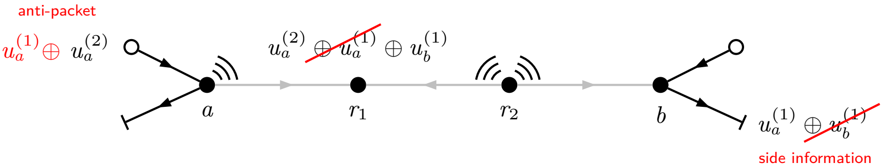
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



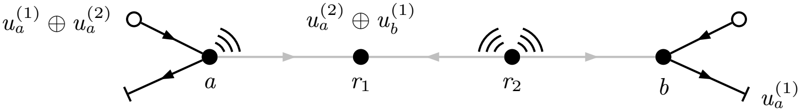
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \dots$ and $u_b^{(1)}, u_b^{(2)}, \dots$



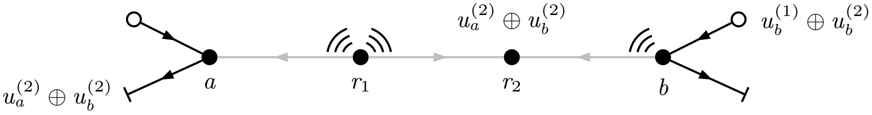
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



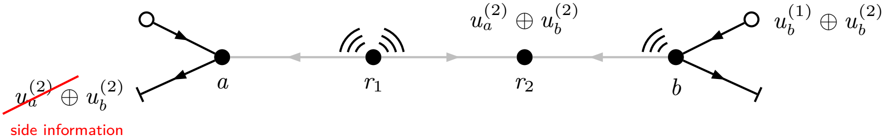
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



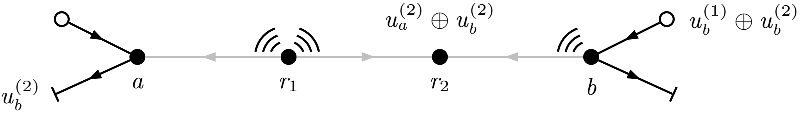
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



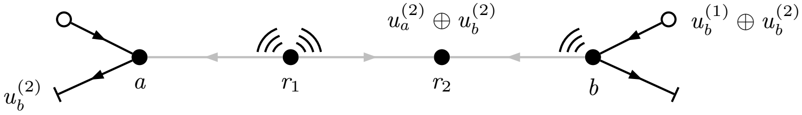
Another Toy Problem

- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



Another Toy Problem

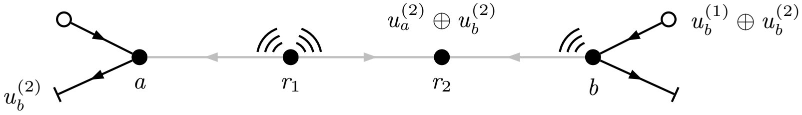
- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



- Steady state

Another Toy Problem

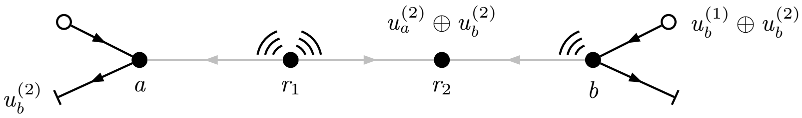
- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



- Steady state
- Both users send a new packet or receive a packet in each block

Another Toy Problem

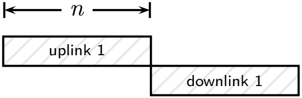
- Noiseless, finite field physical layer and half-duplex nodes
- Users exchange bits (packets) $u_a^{(1)}, u_a^{(2)}, \ldots$ and $u_b^{(1)}, u_b^{(2)}, \ldots$



- Steady state
- Both users send a new packet or receive a packet in each block
- Sum rate approaches 2 bits per 2 transmissions

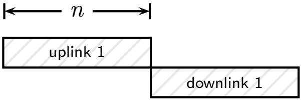
Full-Duplex Block Transmission

Full-Duplex Block Transmission



Two phases:

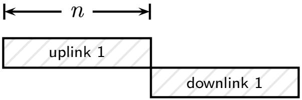
Full-Duplex Block Transmission



Two phases:

1. Uplink: users a and b transmit to the relay r.

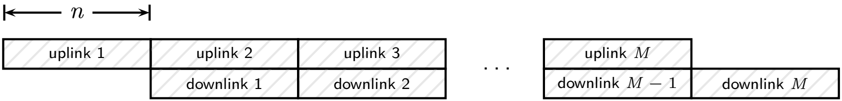
Full-Duplex Block Transmission



Two phases:

1. Uplink: users a and b transmit to the relay r.
2. Downlink: relay r broadcasts to the users a and b.

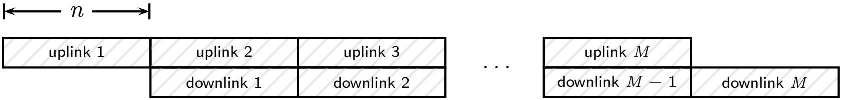
Full-Duplex Block Transmission



Two phases:

1. Uplink: users a and b transmit to the relay r.
2. Downlink: relay r broadcasts to the users a and b.

Full-Duplex Block Transmission



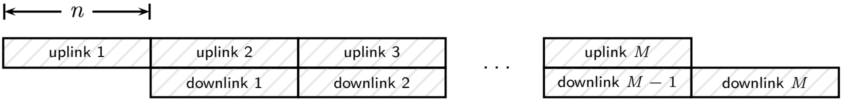
Two phases:

1. Uplink: users a and b transmit to the relay r.
2. Downlink: relay r broadcasts to the users a and b.

Choose number of blocks M (uplink-downlink pairs) large, such that

$$\frac{MnR_a}{(M+1)n} \approx R_a, \qquad \frac{MnR_b}{(M+1)n} \approx R_b.$$

Full-Duplex Block Transmission



Two phases:

1. Uplink: users a and b transmit to the relay r.
2. Downlink: relay r broadcasts to the users a and b.

Choose number of blocks M (uplink-downlink pairs) large, such that

$$\frac{MnR_a}{(M+1)n} \approx R_a, \qquad \frac{MnR_b}{(M+1)n} \approx R_b.$$

What rate pairs (R_a, R_b) (in bits per channel use) are achievable with $p_e \to 0$ as $n \to \infty$?