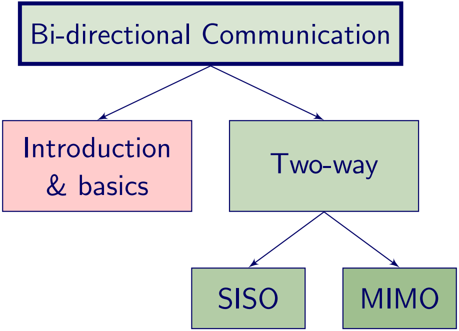
Multi-way Communication and Cooperation

Anas Chaaban* and Aydin Sezgin†

* Computer, Electrical and Mathematical Sciences and Engineering,
KAUST, Thuwal, KSA
† Institute of Digital Communication Systems,
RUB, Bochum, Germany

Structure


Bi-directional Communication
Introduction & basics
Two-way
SISO
MIMO

Part 1: Intro. & Basics

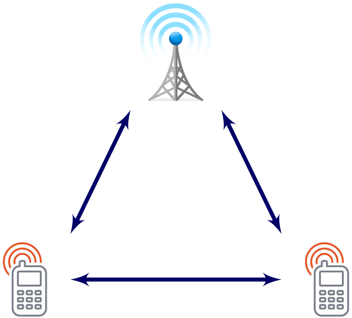
Outline

1. What is bi-directional communication?
2. Why multi-way?
3. History
   - Point-to-point
   - Multiple-access channel
   - Broadcast channel
   - Relay channel

What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

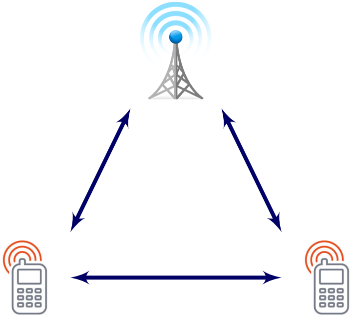


What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device



What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,

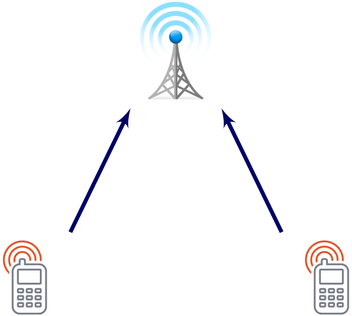
What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,



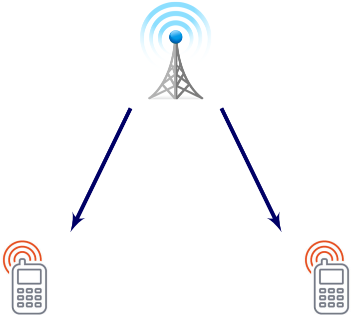
What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,
  - Broadcast channel,



What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,
  - Broadcast channel,
- Bi-directional treatment (two-way):
  - Two-way channel

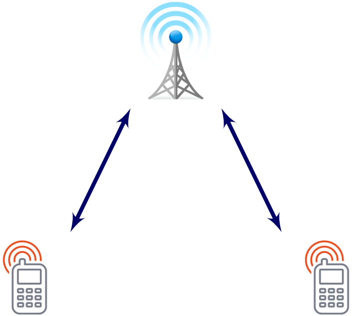
What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,
  - Broadcast channel,
- Bi-directional treatment (two-way):
  - Two-way channel
  - Two-way relay channel



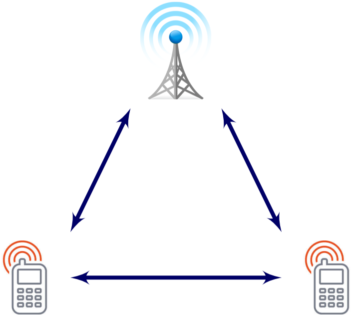
What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,
  - Broadcast channel,
- Bi-directional treatment (two-way):
  - Two-way channel
  - Two-way relay channel
  - multi-way, etc.



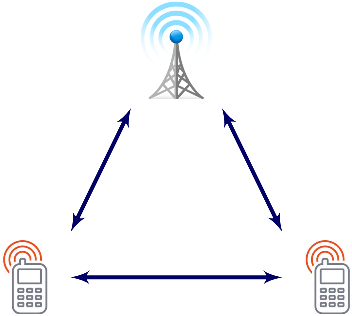
What is bi-directional Communications

Definition

Nodes acting as sources and destinations simultaneously.

Example: Device-to-device

- uni-directional treatment (one way):
  - Point-to-point channel,
  - Multiple-access channel,
  - Broadcast channel,
- Bi-directional treatment (two-way):
  - Two-way channel
  - Two-way relay channel
  - multi-way, etc.



Goal

Introduce and discuss techniques for bi-directional communications.

Outline

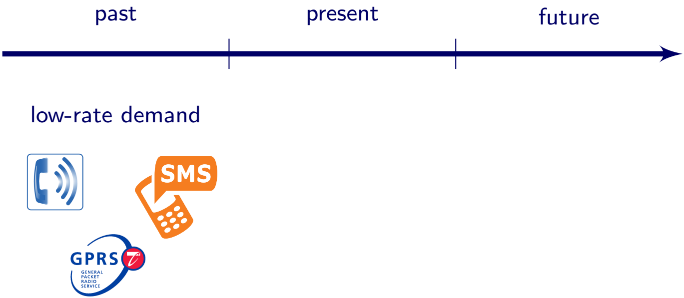
1. What is bi-directional communication?
2. Why multi-way?
3. History
   - Point-to-point
   - Multiple-access channel
   - Broadcast channel
   - Relay channel

Changing Game

Rapid changes in communications: applications, services, requirements, etc.

Changing Game

Rapid changes in communications: applications, services, requirements, etc.



Sources: experian.com, allmytech.pk, en.wikipedia.org, ecnmag.com, kpcb.com, play.google.com

Changing Game

Rapid changes in communications: applications, services, requirements, etc.

past | present | future

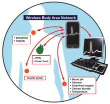
low-rate demand | high-rate demand

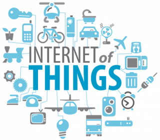
Sources: experian.com, allmytech.pk, en.wikipedia.org, ecnmag.com, kpcb.com, play.google.com

Changing Game

Rapid changes in communications: applications, services, requirements, etc.

past	present	future
low-rate demand	high-rate demand	???

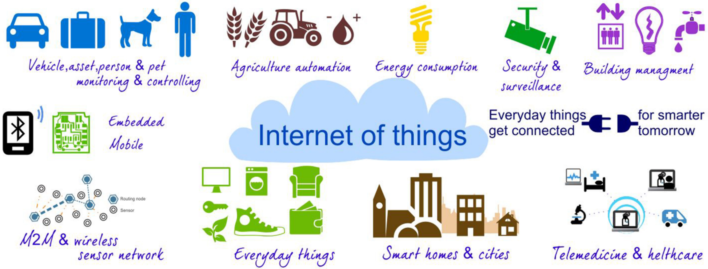




Sources: experian.com, allmytech.pk, en.wikipedia.org, ecnmag.com, kpcb.com, play.google.com

Changing Game

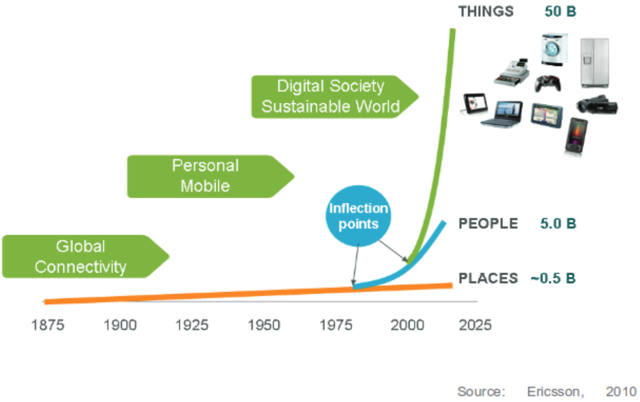
Future: Everything can communicate!



New players, new rules!

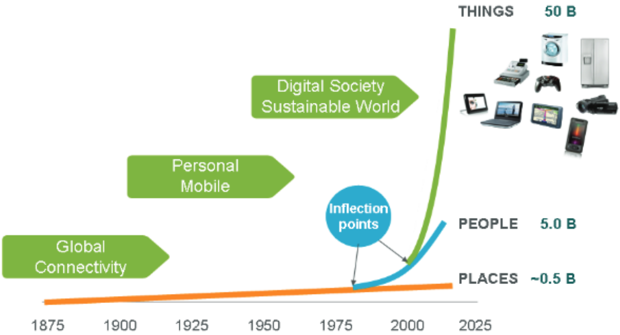
Sources: influxis.com

Towards 50B devices in 2020!



Increasing number of connected devices (IoT, M2M, etc.)

Towards 50B devices in 2020!

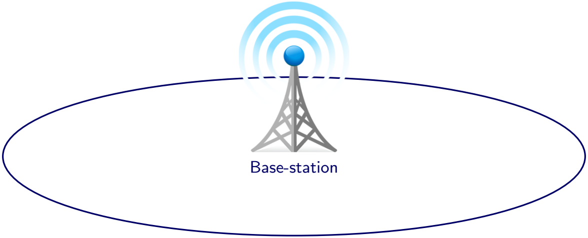


Source: Ericsson, 2010

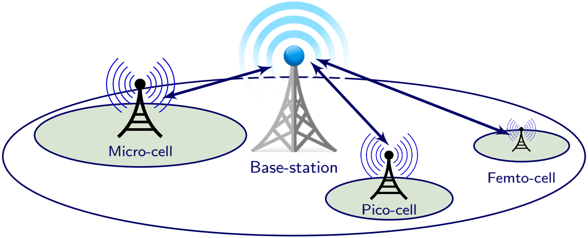
Increasing number of connected devices (IoT, M2M, etc.)

Consequence: Networks must support much higher data-rates

Ideas

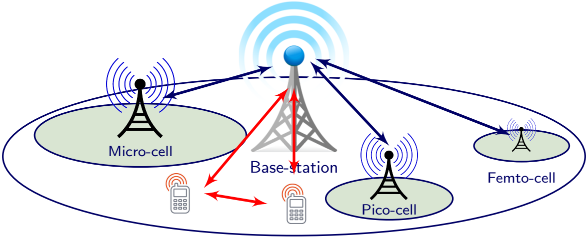


Ideas



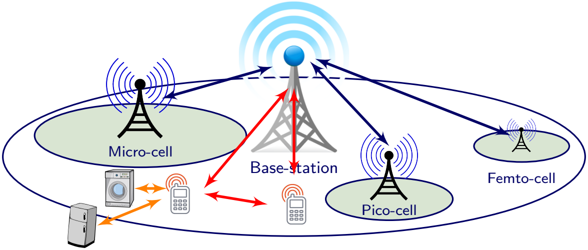
- Densification of networks

Ideas



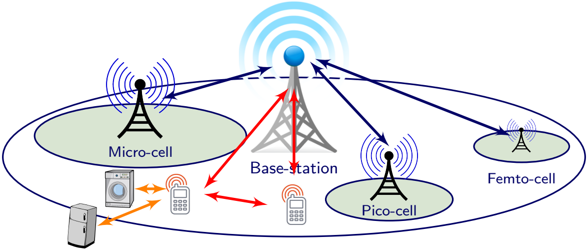
- Densification of networks
- Device-to-device

Ideas



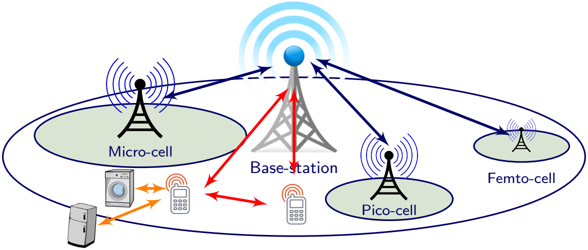
- Densification of networks
- Device-to-device
- IoT, etc.

Ideas



- Densification of networks
- Device-to-device
- IoT, etc.
- Important: Most communication is bi-directional (uplink/downlink, feedback, etc.)

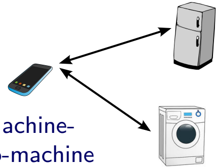
Ideas



- Densification of networks
- Device-to-device
- IoT, etc.
- Important: Most communication is bi-directional (uplink/downlink, feedback, etc.)

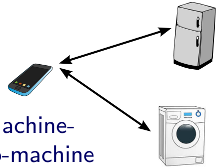
$\Rightarrow$ need to study bi-directional communication

More sophisticated network topologies!

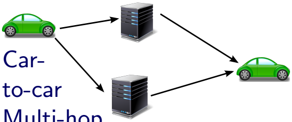


Machine-
to-machine
Point↔multi-point

More sophisticated network topologies!

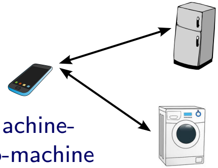


Machine-
to-machine
Point↔multi-point

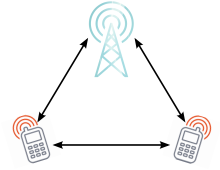


Car-
to-car
Multi-hop

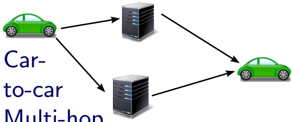
More sophisticated network topologies!



Machine-to-machine
Point↔multi-point

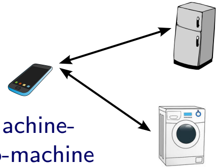


Device-to-device

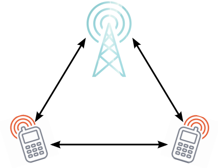


Car-to-car
Multi-hop

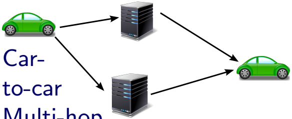
More sophisticated network topologies!



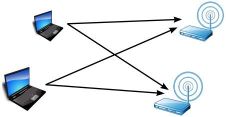
Machine-to-machine
Point↔multi-point



Device-to-device

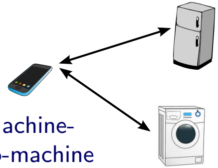


Car-to-car
Multi-hop

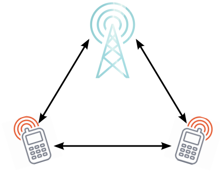


X-Channel

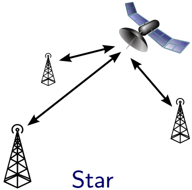
More sophisticated network topologies!



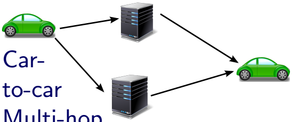
Machine-
to-machine
Point↔multi-point



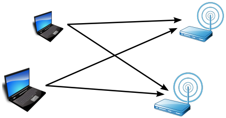
Device-to-device



Star

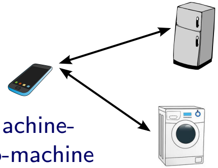


Car-
to-car
Multi-hop

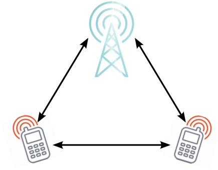


X-Channel

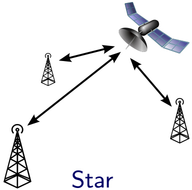
More sophisticated network topologies!



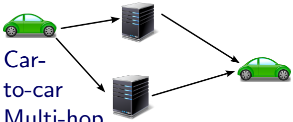
Machine-
to-machine
Point↔multi-point



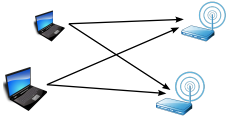
Device-to-device



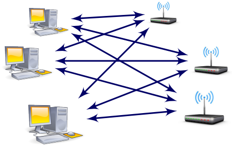
Star



Car-
to-car
Multi-hop

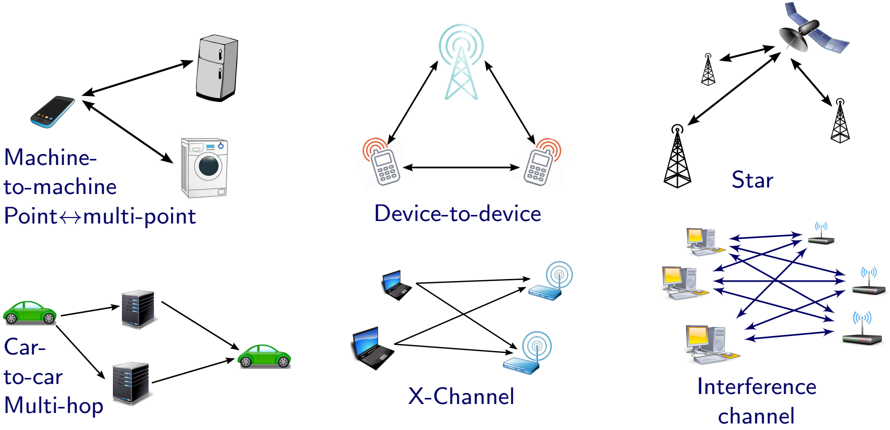


X-Channel



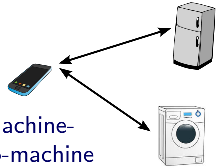
Interference
channel

More sophisticated network topologies!

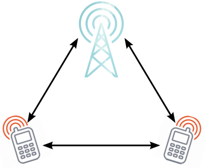


- Important factors: Multi-way communications and Relaying!

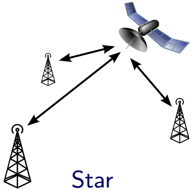
More sophisticated network topologies!



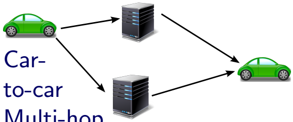
Machine-to-machine
Point↔multi-point



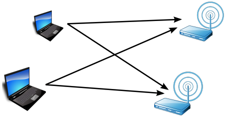
Device-to-device



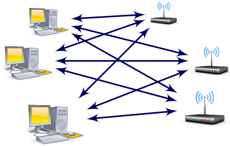
Star



Car-to-car
Multi-hop



X-Channel

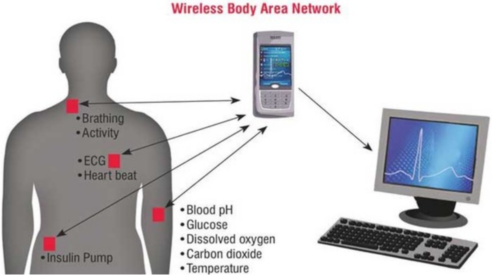


Interference channel

- Important factors: Multi-way communications and Relaying!

⇒ need to study bi-directional communication

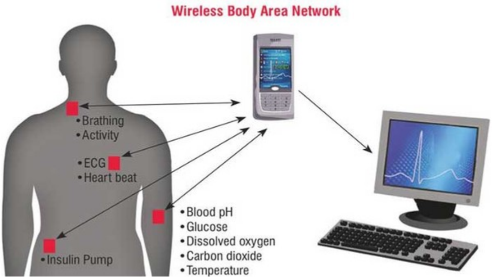
Body-area networks



- Multiple sensors communicating with a central node,

Sources: www.examiner.com

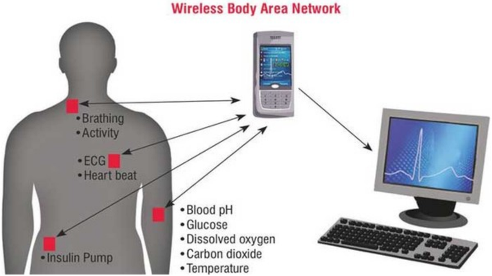
Body-area networks



- Multiple sensors communicating with a central node,
- Higher spectral/power efficiency $\Rightarrow$ shorter transmission duration $\Rightarrow$ longer life-cycle/less radiation

Sources: www.examiner.com

Body-area networks



- Multiple sensors communicating with a central node,
- Higher spectral/power efficiency ⇒ shorter transmission duration ⇒ longer life-cycle/less radiation

⇒ need to study bi-directional communication

Sources: www.examiner.com

Tutorial Topics

- Two-way channel,

Tutorial Topics

- Two-way channel,

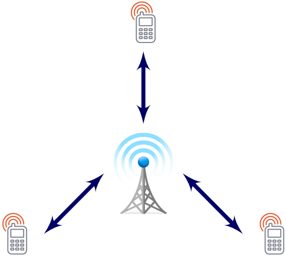
- Two-way relay channel,

Tutorial Topics

- Two-way channel,

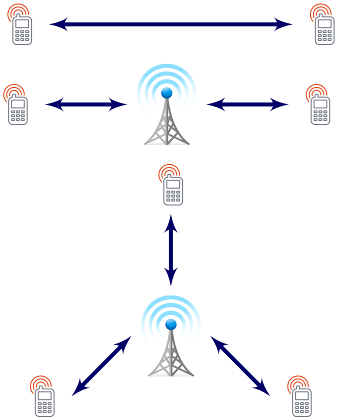
- Two-way relay channel,

- Multi-way relay channel,



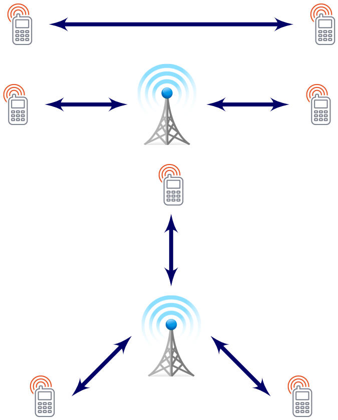
Tutorial Topics

- Two-way channel,
- Two-way relay channel,
- Multi-way relay channel,
- MIMO cases,



Tutorial Topics

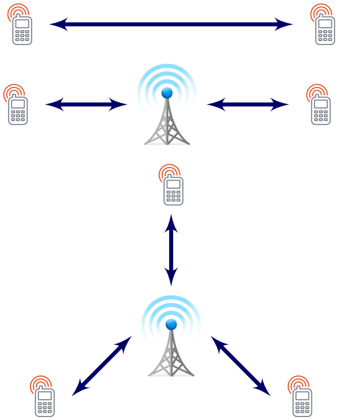
- Two-way channel,
- Two-way relay channel,
- Multi-way relay channel,
- MIMO cases,



One-way communication is an integral part of two-way communication

Tutorial Topics

- Two-way channel,
- Two-way relay channel,
- Multi-way relay channel,
- MIMO cases,



One-way communication is an integral part of two-way communication

Brief Review

Outline

1. What is bi-directional communication?

2. Why multi-way?

3. History
   - Point-to-point
   - Multiple-access channel
   - Broadcast channel
   - Relay channel

History

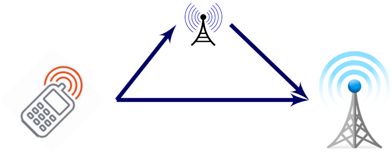
In the beginning... One-way (uni-directional) Communications:

- Point-to-point (P2P): [Shannon 48],

History

In the beginning... One-way (uni-directional) Communications:

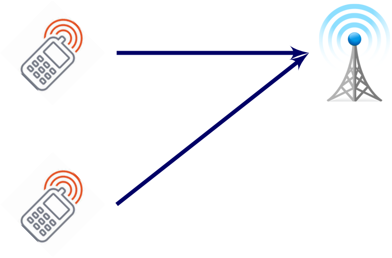
- Point-to-point (P2P): [Shannon 48],
- Relay channel (RC): [van der Meulen 71], [Cover & El-Gamal 79],



History

In the beginning... One-way (uni-directional) Communications:

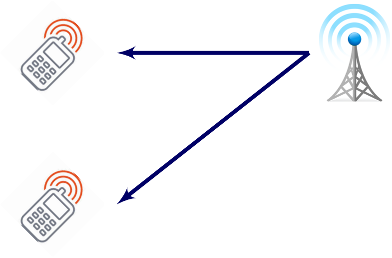
- Point-to-point (P2P): [Shannon 48],
- Relay channel (RC): [van der Meulen 71], [Cover & El-Gamal 79],
- Multiple-access channel (MAC): [Ahlswede 71], [Liao 72],



History

In the beginning... One-way (uni-directional) Communications:

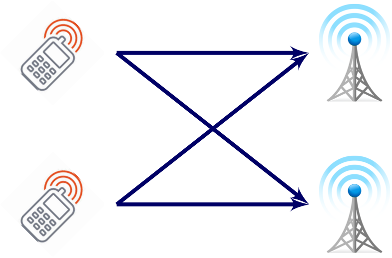
- Point-to-point (P2P): [Shannon 48],
- Relay channel (RC): [van der Meulen 71], [Cover & El-Gamal 79],
- Multiple-access channel (MAC): [Ahlswede 71], [Liao 72],
- Broad-cast channel (BC): [Cover 72], [Bergmans 73],



History

In the beginning... One-way (uni-directional) Communications:

- Point-to-point (P2P): [Shannon 48],
- Relay channel (RC): [van der Meulen 71], [Cover & El-Gamal 79],
- Multiple-access channel (MAC): [Ahlswede 71], [Liao 72],
- Broad-cast channel (BC): [Cover 72], [Bergmans 73],
- Interference Channel (IC): [Carleial 75], [Han & Koayashi 81], [Sato 81],



P2P

One transmitter, one receiver:

P2P

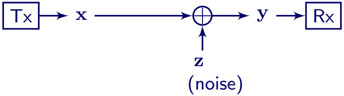
One transmitter, one receiver:

- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,

P2P

One transmitter, one receiver:

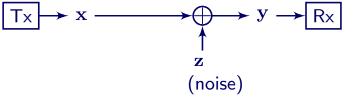
- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,
- Output: $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is noise,

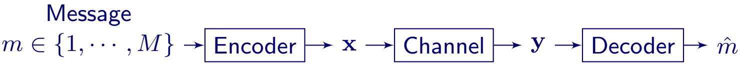


P2P

One transmitter, one receiver:

- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,
- Output: $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is noise,

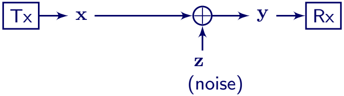




P2P

One transmitter, one receiver:

- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,
- Output: $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is noise,



Message

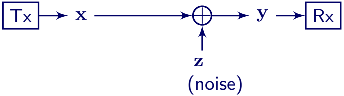
$m \in \{1, \cdots, M\}$ → Encoder → $\mathbf{x}$ → Channel → $\mathbf{y}$ → Decoder → $\hat{m}$

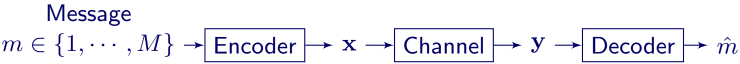
- Error probability: $P_e = \text{Prob}(\hat{m} \neq m)$,

P2P

One transmitter, one receiver:

- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,
- Output: $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is noise,



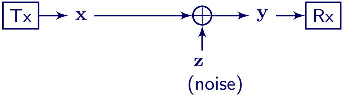


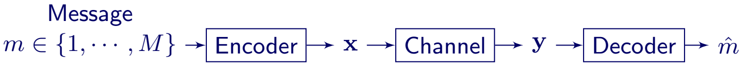
- Error probability: $P_e = \text{Prob}(\hat{m} \neq m)$,
- Reliable Comm: $P_e \to 0$ as $n \to \infty$,

P2P

One transmitter, one receiver:

- Input: $\mathbf{x} = (x(1), \cdots, x(n))$ with power P,
- Output: $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is noise,





- Error probability: $P_e = \text{Prob}(\hat{m} \neq m)$,
- Reliable Comm: $P_e \to 0$ as $n \to \infty$,
- Goal: Find the maximum M.

P2P: Simple example

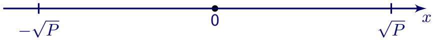
Let the code-length $n = 1$

P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$

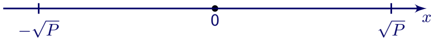


P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$

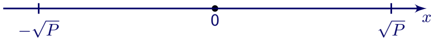


P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$
- Question: How many points can we place between $-\sqrt{P}$ and $\sqrt{P}$ at a distance of 2σ?

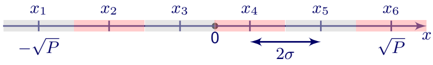


P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$
- Question: How many points can we place between $-\sqrt{P}$ and $\sqrt{P}$ at a distance of 2σ?



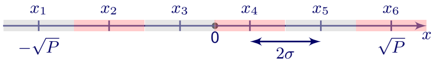
- Answer: $M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}}$

P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$
- Question: How many points can we place between $-\sqrt{P}$ and $\sqrt{P}$ at a distance of 2σ?



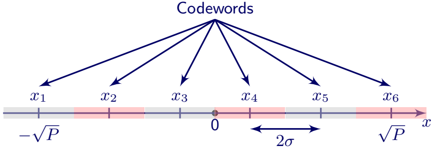
- Answer: $M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}}$ (1-D sphere-packing)

P2P: Simple example

Let the code-length $n = 1$

Assume:

- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$
- Question: How many points can we place between $-\sqrt{P}$ and $\sqrt{P}$ at a distance of 2σ?



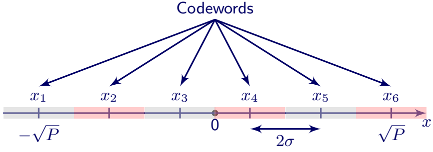
- Answer: $M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}}$ (1-D sphere-packing)
- M codewords $\Rightarrow \log_2(M) = \frac{1}{2}\log_2(\text{SNR})$ bits,

P2P: Simple example

Let the code-length $n = 1$

Assume:

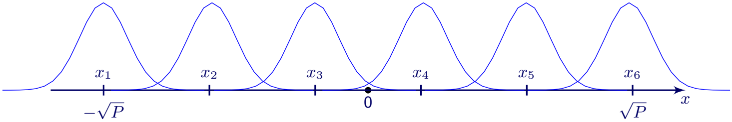
- Tx power P, e.g. $|x| < \sqrt{P}$
- Noise $|z| < \sigma$
- Question: How many points can we place between $-\sqrt{P}$ and $\sqrt{P}$ at a distance of 2σ?



- Answer: $M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}}$ (1-D sphere-packing)
- M codewords $\Rightarrow \log_2(M) = \frac{1}{2}\log_2(\text{SNR})$ bits,
- Rate $R = \frac{\log_2(M)}{n} = \frac{1}{2}\log_2(\text{SNR})$ bits per transmission.

P2P

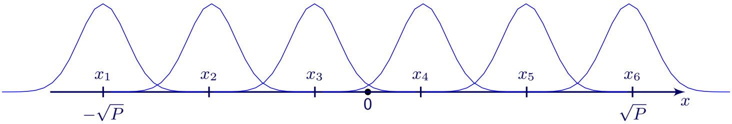
In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).



- With $n = 1$, decoding errors occur,

P2P

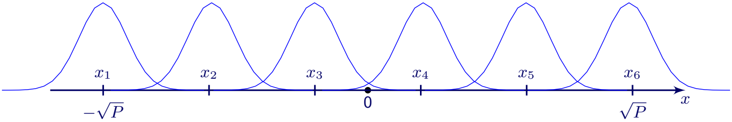
In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).



- With $n = 1$, decoding errors occur,
- instead, encoding with $n \gg 1$,

P2P

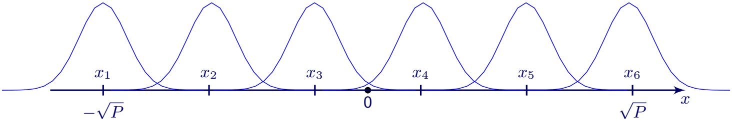
In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).



- With $n = 1$, decoding errors occur,
- instead, encoding with $n \gg 1$,
- number of codewords: $M = (\sqrt{1 + \text{SNR}})^n$ (n-D sphere-packing),

P2P

In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).

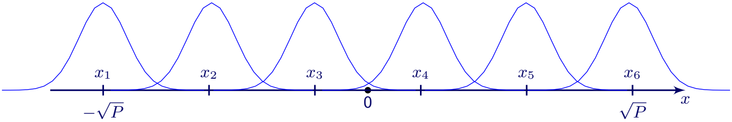


- With $n = 1$, decoding errors occur,
- instead, encoding with $n \gg 1$,
- number of codewords: $M = (\sqrt{1 + \text{SNR}})^n$ (n-D sphere-packing),

$\Rightarrow$ Rate $= \frac{\log_2(M)}{n} = \frac{1}{2}\log_2(1 + \text{SNR})$ bits per transmission,

P2P

In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).



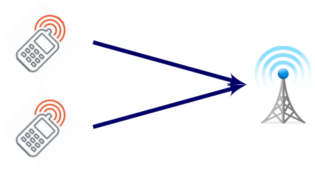
- With $n = 1$, decoding errors occur,
- instead, encoding with $n \gg 1$,
- number of codewords: $M = (\sqrt{1 + \text{SNR}})^n$ (n-D sphere-packing),

$\Rightarrow$ Rate $= \frac{\log_2(M)}{n} = \frac{1}{2}\log_2(1 + \text{SNR})$ bits per transmission,

- Capacity [Shannon 48].

MAC

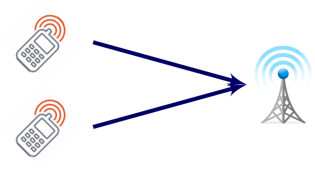
Two transmitters, one receiver:



MAC

Two transmitters, one receiver:

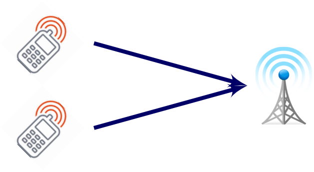
- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Powers P_1 and P_2,

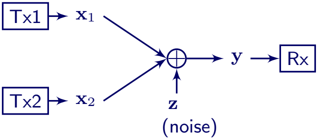


MAC

Two transmitters, one receiver:

- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Powers P_1 and P_2,
- Output: $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$,

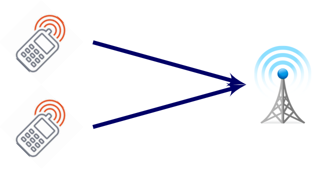


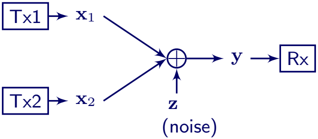


MAC

Two transmitters, one receiver:

- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Powers P_1 and P_2,
- Output: $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$,
- Goal: Find the rate-region.

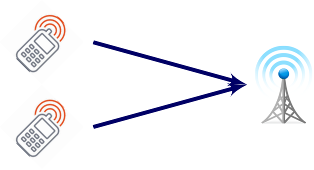


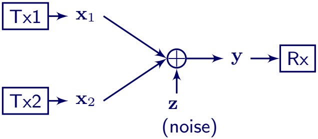


MAC

Two transmitters, one receiver:

- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Powers P_1 and P_2,
- Output: $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$,
- Goal: Find the rate-region.





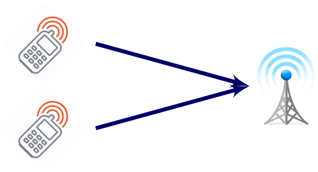
Rate-region

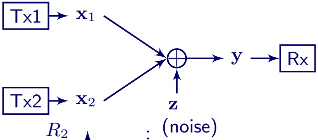
The rate-region is the set of achievable rate pairs (R_1, R_2).

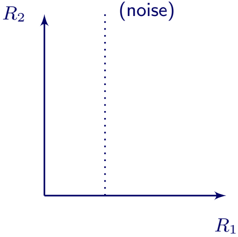
MAC

- Successive decoding:
- Treat $\mathbf{x}_2$ as noise $\Rightarrow$

$$R_1 = \frac{1}{2}\log\left(1 + \frac{P_1}{\sigma^2 + P_2}\right),$$







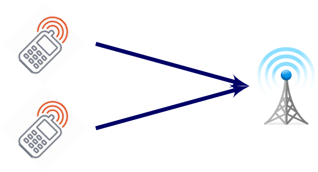
MAC

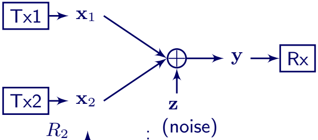
- Successive decoding:
- Treat $\mathbf{x}_2$ as noise $\Rightarrow$

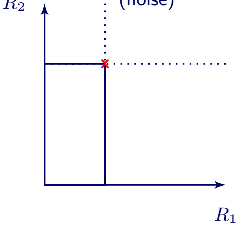
$$R_1 = \frac{1}{2} \log \left(1 + \frac{P_1}{\sigma^2 + P_2} \right),$$

- Subtract $\mathbf{x}_1$ and decode $\mathbf{x}_2$ $\Rightarrow$

$$R_2 = \frac{1}{2} \log \left(1 + \frac{P_2}{\sigma^2} \right),$$







MAC

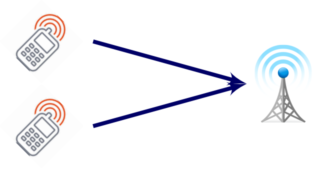
- Successive decoding:
- Treat $\mathbf{x}_2$ as noise $\Rightarrow$

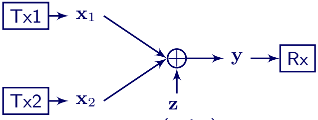
$$R_1 = \frac{1}{2}\log\left(1 + \frac{P_1}{\sigma^2 + P_2}\right),$$

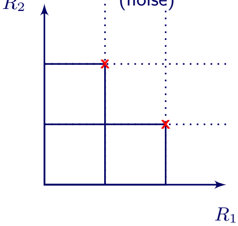
- Subtract $\mathbf{x}_1$ and decode $\mathbf{x}_2$ $\Rightarrow$

$$R_2 = \frac{1}{2}\log\left(1 + \frac{P_2}{\sigma^2}\right),$$

- opposite order,







MAC

- Successive decoding:
- Treat $\mathbf{x}_2$ as noise $\Rightarrow$

$$R_1 = \frac{1}{2}\log\left(1 + \frac{P_1}{\sigma^2 + P_2}\right),$$

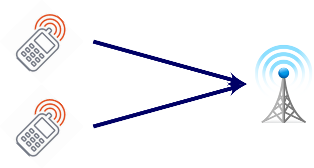
- Subtract $\mathbf{x}_1$ and decode $\mathbf{x}_2$ $\Rightarrow$

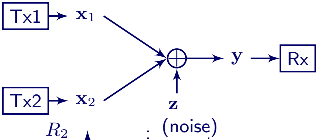
$$R_2 = \frac{1}{2}\log\left(1 + \frac{P_2}{\sigma^2}\right),$$

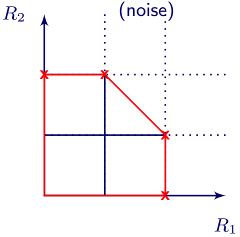
- opposite order,
- Capacity region [Ahlswede 71],

$$R_i \leq \frac{1}{2}\log\left(1 + \frac{P_i}{\sigma^2}\right), \quad i = 1, 2,$$

$$R_1 + R_2 \leq \frac{1}{2}\log\left(1 + \frac{P_1 + P_2}{\sigma^2}\right).$$

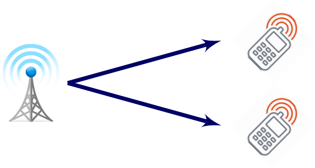






BC

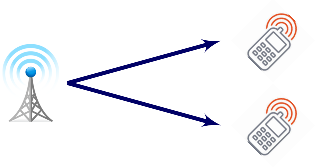
one transmitter, two receivers:



BC

one transmitter, two receivers:

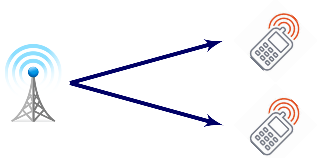
- Input: $\mathbf{x}$ with power P,

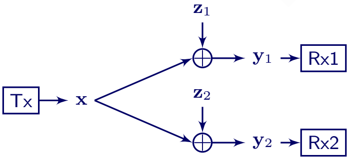


BC

one transmitter, two receivers:

- Input: $\mathbf{x}$ with power P,
- Outputs: $\mathbf{y}_i = \mathbf{x} + \mathbf{z}_i$,
- Noises: $\sigma_1^2 \leq \sigma_2^2$,

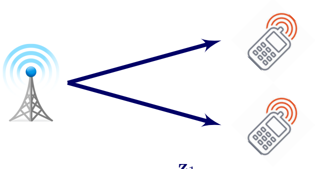


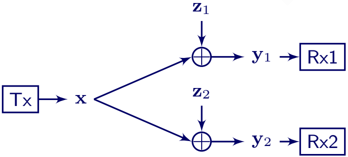


BC

one transmitter, two receivers:

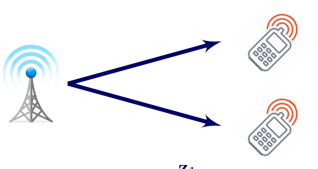
- Input: $\mathbf{x}$ with power P,
- Outputs: $\mathbf{y}_i = \mathbf{x} + \mathbf{z}_i$,
- Noises: $\sigma_1^2 \leq \sigma_2^2$,
- Goal: Find the rate-region,

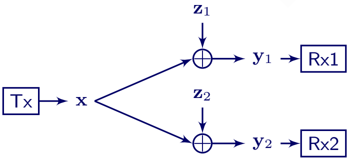




BC

- Superposition coding:
- Send $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, powers $P_1 + P_2 = P$,

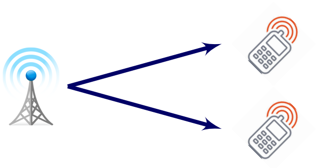


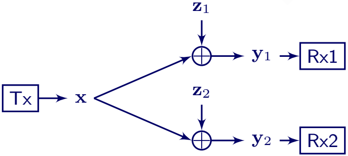


BC

- Superposition coding:
- Send $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, powers $P_1 + P_2 = P$,
- Both receivers decode $\mathbf{x}_2$ treating $\mathbf{x}_1$ as noise $\Rightarrow$

$$R_2 = \frac{1}{2} \log \left(1 + \frac{P_2}{\sigma_2^2 + P_1} \right),$$





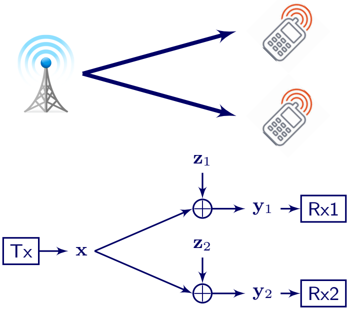
BC

- Superposition coding:
- Send $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, powers $P_1 + P_2 = P$,
- Both receivers decode $\mathbf{x}_2$ treating $\mathbf{x}_1$ as noise $\Rightarrow$

$$R_2 = \frac{1}{2}\log\left(1 + \frac{P_2}{\sigma_2^2 + P_1}\right),$$

- Rx1 subtracts $\mathbf{x}_2$ and decodes $\mathbf{x}_1$ $\Rightarrow$

$$R_1 = \frac{1}{2}\log\left(1 + \frac{P_1}{\sigma_1^2}\right),$$



BC

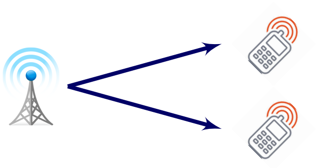
- Superposition coding:
- Send $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, powers $P_1 + P_2 = P$,
- Both receivers decode $\mathbf{x}_2$ treating $\mathbf{x}_1$ as noise $\Rightarrow$

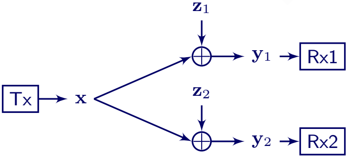
$$R_2 = \frac{1}{2}\log\left(1 + \frac{P_2}{\sigma_2^2 + P_1}\right),$$

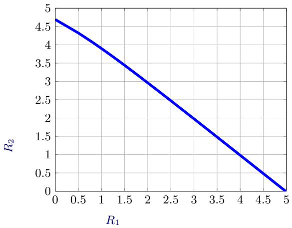
- Rx1 subtracts $\mathbf{x}_2$ and decodes $\mathbf{x}_1 \Rightarrow$

$$R_1 = \frac{1}{2}\log\left(1 + \frac{P_1}{\sigma_1^2}\right),$$

- Rate-region = Capacity region [Cover 72], [Bergmans 74],

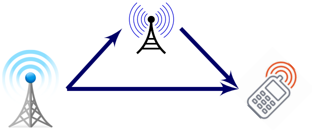






Relay Channel

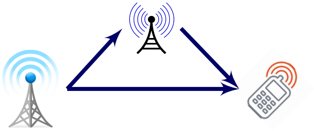
one transmitter, one receiver, one relay:



Relay Channel

one transmitter, one receiver, one relay:

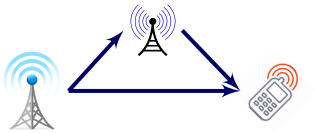
- Inputs: $\mathbf{x}$, $\mathbf{x}_r$,
- Powers: P, P_r,

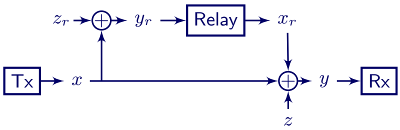


Relay Channel

one transmitter, one receiver, one relay:

- Inputs: $\mathbf{x}$, $\mathbf{x}_r$,
- Powers: P, P_r,
- Outputs: $\mathbf{y} = \mathbf{x} + \mathbf{x}_r + \mathbf{z}$, $\mathbf{y}_r = \mathbf{x} + \mathbf{z}_r$,

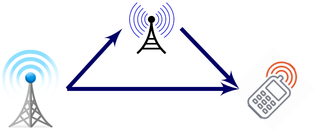


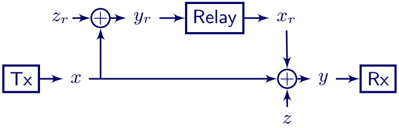


Relay Channel

one transmitter, one receiver, one relay:

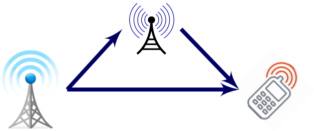
- Inputs: $\mathbf{x}$, $\mathbf{x}_r$,
- Powers: P, P_r,
- Outputs: $\mathbf{y} = \mathbf{x} + \mathbf{x}_r + \mathbf{z}$, $\mathbf{y}_r = \mathbf{x} + \mathbf{z}_r$,
- Noise variance: σ^2 and σ_r^2.

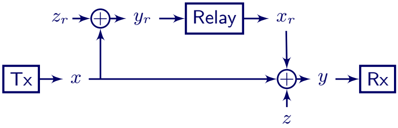




Relay Channel

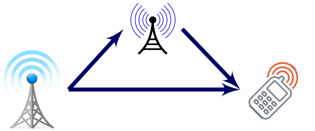
- Main schemes:
  Decode-forward,
  Compress-forward

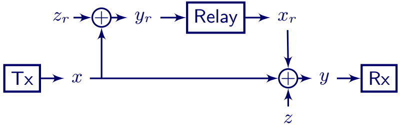




Relay Channel

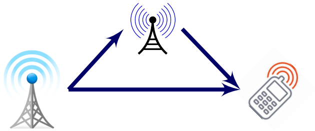
- Main schemes: Decode-forward, Compress-forward
- Notice: Interaction between consecutive symbols/codewords at Rx!

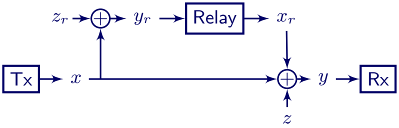




Relay Channel

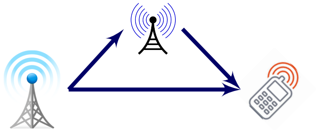
- Main schemes:
  Decode-forward,
  Compress-forward
- Notice: Interaction between consecutive symbols/codewords at Rx!
- Interaction can be exploited using block-Markov encoding [Cover & El-Gamal 79],

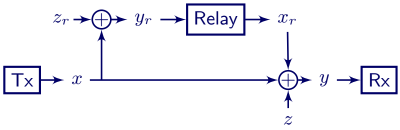


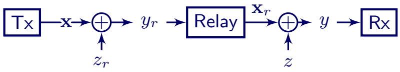


Relay Channel

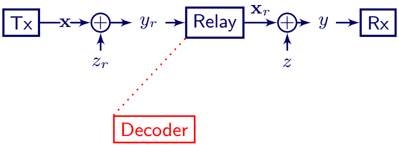
- Main schemes:
  Decode-forward,
  Compress-forward
- Notice: Interaction between consecutive symbols/codewords at Rx!
- Interaction can be exploited using block-Markov encoding [Cover & El-Gamal 79],
- For simplicity: Consider a separated relay channel







Relay Channel: DF

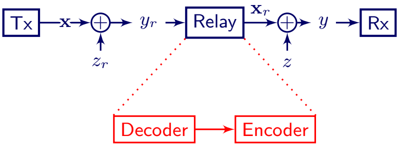


Decode-forward:

- Relay decodes $\mathbf{x}$ $\Rightarrow$

$$R \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right).$$

Relay Channel: DF



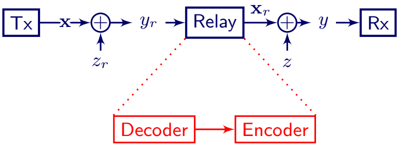
Decode-forward:

- Relay decodes $\mathbf{x} \Rightarrow$

$$R \leq \frac{1}{2} \log \left(1 + \frac{P}{\sigma_r^2} \right).$$

- Relay re-encodes $\mathbf{x}$ to $\mathbf{x}_r$ and sends it.

Relay Channel: DF



Decode-forward:

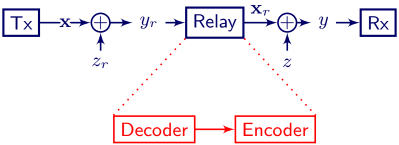
- Relay decodes $\mathbf{x}$ $\Rightarrow$

$$R \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right).$$

- Relay re-encodes $\mathbf{x}$ to $\mathbf{x}_r$ and sends it.
- Rx decodes x_r $\Rightarrow$

$$R \leq \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right).$$

Relay Channel: DF



Decode-forward:

- Relay decodes $\mathbf{x}$ $\Rightarrow$

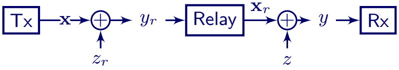
$$R \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right).$$

- Relay re-encodes $\mathbf{x}$ to $\mathbf{x}_r$ and sends it.
- Rx decodes x_r $\Rightarrow$

$$R \leq \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right).$$

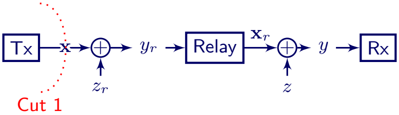
- Achievable rate $\min\left\{\frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right), \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)\right\}$.

Relay Channel: Cut-set bound



Cut-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

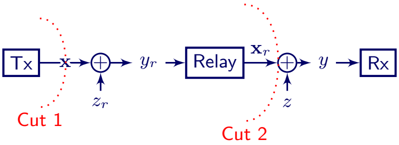
Relay Channel: Cut-set bound



Cut-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

- Cut 1 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right)$.

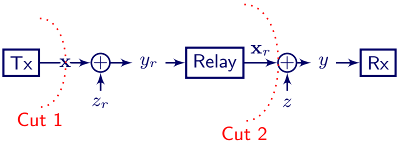
Relay Channel: Cut-set bound



Cut-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

- Cut 1 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right)$.
- Cut 2 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)$.

Relay Channel: Cut-set bound

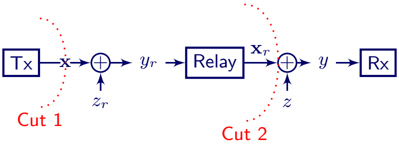


Cut-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

- Cut 1 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right)$.
- Cut 2 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)$.

$\Rightarrow$ Upper bound $C \leq \min\left\{\frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right), \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)\right\}$.

Relay Channel: Cut-set bound



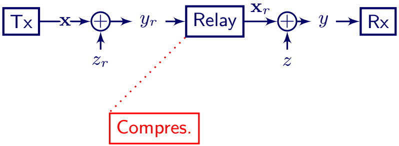
Cut-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

- Cut 1 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right)$.
- Cut 2 $\Rightarrow C \leq \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)$.

$\Rightarrow$ Upper bound $C \leq \min\left\{\frac{1}{2}\log\left(1+\frac{P}{\sigma_r^2}\right), \frac{1}{2}\log\left(1+\frac{P_r}{\sigma^2}\right)\right\}$.

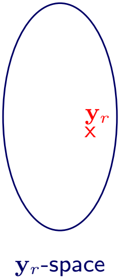
- DF Optimal: Coincides with the cut-set bound.

Relay Channel: CF

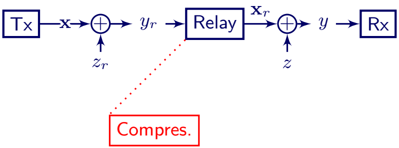


Compress-forward:

- Relay compresses/quantizes $\mathbf{y}_r$ to $\hat{\mathbf{y}}_r = \mathbf{y}_r + \mathbf{z}_c$ ($\mathbf{z}_c$ compression noise)

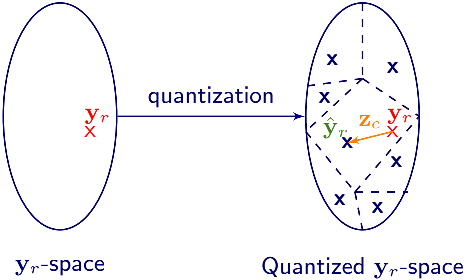


Relay Channel: CF

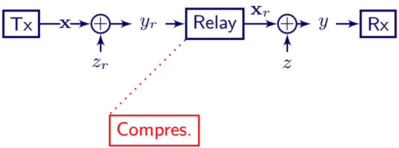


Compress-forward:

- Relay compresses/quantizes $\mathbf{y}_r$ to $\hat{\mathbf{y}}_r = \mathbf{y}_r + \mathbf{z}_c$ ($\mathbf{z}_c$ compression noise)

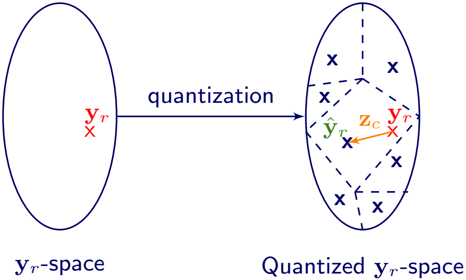


Relay Channel: CF

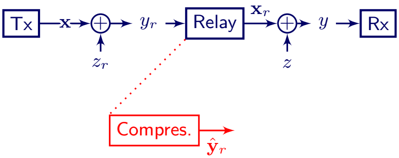


Compress-forward:

- Relay compresses/quantizes $\mathbf{y}_r$ to $\hat{\mathbf{y}}_r = \mathbf{y}_r + \mathbf{z}_c$ ($\mathbf{z}_c$ compression noise)
- Compression rate R_c ($\frac{1}{n}\log$(number of bins))



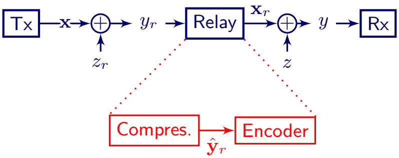
Relay Channel: CF



Compress-forward:

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)

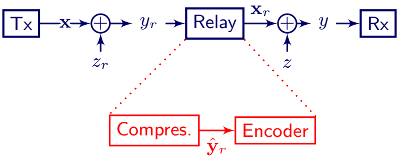
Relay Channel: CF



Compress-forward:

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)
- Relay encodes $\hat{\mathbf{y}}_r$ to $\mathbf{x}_r$ (rate R_c).

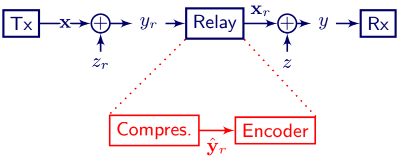
Relay Channel: CF



Compress-forward:

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)
- Relay encodes $\hat{\mathbf{y}}_r$ to $\mathbf{x}_r$ (rate R_c).
- Rx decodes $\mathbf{x}_r \Rightarrow R_c = \frac{1}{2} \log \left(1 + \frac{P_r}{\sigma^2}\right)$.

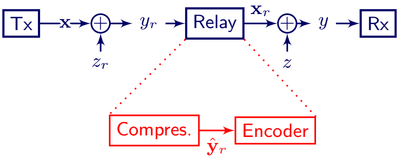
Relay Channel: CF



Compress-forward:

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)
- Relay encodes $\hat{\mathbf{y}}_r$ to $\mathbf{x}_r$ (rate R_c).
- Rx decodes $\mathbf{x}_r \Rightarrow R_c = \frac{1}{2}\log\left(1 + \frac{P_r}{\sigma^2}\right)$.
- Rx obtains $\hat{\mathbf{y}}_r = \mathbf{x} + \mathbf{z}_r + \mathbf{z}_c$.

Relay Channel: CF



Compress-forward:

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)
- Relay encodes $\hat{\mathbf{y}}_r$ to $\mathbf{x}_r$ (rate R_c).
- Rx decodes $\mathbf{x}_r \Rightarrow R_c = \frac{1}{2}\log\left(1 + \frac{P_r}{\sigma^2}\right)$.
- Rx obtains $\hat{\mathbf{y}}_r = \mathbf{x} + \mathbf{z}_r + \mathbf{z}_c$.
- Rx then decodes $\mathbf{x}$ from $\hat{\mathbf{y}}_r \Rightarrow$
  $R = \frac{1}{2}\log\left(1 + \frac{P}{\sigma_r^2 + D}\right) \leq \frac{1}{2}\log\left(1 + \frac{P(P_r+\sigma^2)}{\sigma_r^2(P_r+\sigma^2)+\sigma^2(P+\sigma_r^2)}\right)$.

Part 2: SISO Bi-directional

Outline

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

- Outer bound: Cut-set $R_i \leq I(X_i, Y_j | X_j)$ maximized over $P(X_1, X_2)$,

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

- Outer bound: Cut-set $R_i \leq I(X_i, Y_j | X_j)$ maximized over $P(X_1, X_2)$,
- Inner bound: P2P codebooks $R_i \leq I(X_i, Y_j | X_j)$ maximized over $P(X_1, X_2) = P(X_1)P(X_2)$,

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

- Outer bound: Cut-set $R_i \leq I(X_i, Y_j|X_j)$ maximized over $P(X_1, X_2)$,
- Inner bound: P2P codebooks $R_i \leq I(X_i, Y_j|X_j)$ maximized over $P(X_1, X_2) = P(X_1)P(X_2)$,
- $P(X_1, X_2)$ vs. $P(X_1)P(X_2)$:
  - $P(X_1, X_2)$ allows interactive coding: X_i and Y_i can be dependent, $X_1 = \mathcal{E}(m_1, Y_1)$
  - $P(X_1)P(X_2)$ does not allow interactive coding: X_i and Y_i independent, $X_1 = \mathcal{E}(m_1)$

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

- Outer bound: Cut-set $R_i \leq I(X_i, Y_j|X_j)$ maximized over $P(X_1, X_2)$,
- Inner bound: P2P codebooks $R_i \leq I(X_i, Y_j|X_j)$ maximized over $P(X_1, X_2) = P(X_1)P(X_2)$,
- $P(X_1, X_2)$ vs. $P(X_1)P(X_2)$:
  - $P(X_1, X_2)$ allows interactive coding: X_i and Y_i can be dependent, $X_1 = \mathcal{E}(m_1, Y_1)$
  - $P(X_1)P(X_2)$ does not allow interactive coding: X_i and Y_i independent, $X_1 = \mathcal{E}(m_1)$
- Consequence: Bounds do not coincide

Two-way Channel

Channel with two transceivers: First studied by Shannon (1961)

- Outer bound: Cut-set $R_i \leq I(X_i, Y_j | X_j)$ maximized over $P(X_1, X_2)$,
- Inner bound: P2P codebooks $R_i \leq I(X_i, Y_j | X_j)$ maximized over $P(X_1, X_2) = P(X_1)P(X_2)$,
- $P(X_1, X_2)$ vs. $P(X_1)P(X_2)$:
  - $P(X_1, X_2)$ allows interactive coding: X_i and Y_i can be dependent, $X_1 = \mathcal{E}(m_1, Y_1)$
  - $P(X_1)P(X_2)$ does not allow interactive coding: X_i and Y_i independent, $X_1 = \mathcal{E}(m_1)$
- Consequence: Bounds do not coincide

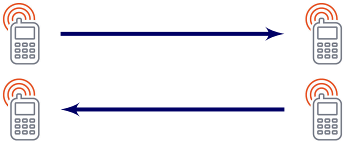
$\Rightarrow$ unknown capacity!

Two-way Channel

- However: Bounds coincide if channel is separable!

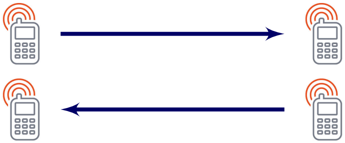
Two-way Channel

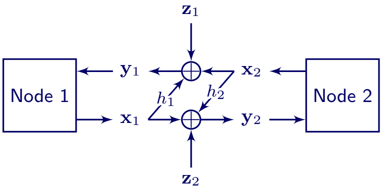
- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.



Two-way Channel

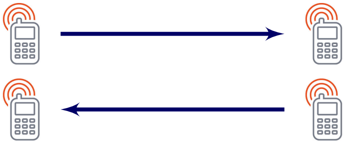
- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.
- The Gaussian channel belongs to this class.

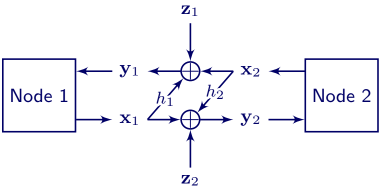




Two-way Channel

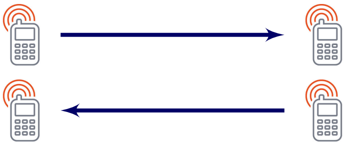
- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.
- The Gaussian channel belongs to this class.
- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$ with powers P_1 and P_2,

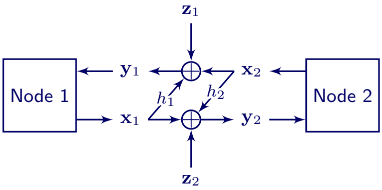




Two-way Channel

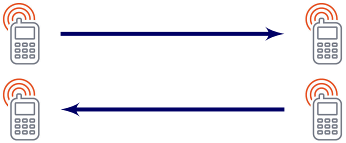
- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.
- The Gaussian channel belongs to this class.
- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$ with powers P_1 and P_2,
- Outputs: $\mathbf{y}_i = \mathbf{x}_j + h_i \mathbf{x}_j + \mathbf{z}_i$, $h_i \in \mathbb{R}$, $\mathbf{z}_i$ has variance σ_i^2,

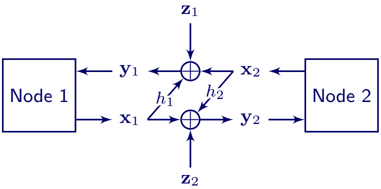


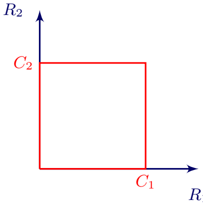


Two-way Channel

- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.
- The Gaussian channel belongs to this class.
- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$ with powers P_1 and P_2,
- Outputs: $\mathbf{y}_i = \mathbf{x}_j + h_i \mathbf{x}_j + \mathbf{z}_i$, $h_i \in \mathbb{R}$, $\mathbf{z}_i$ has variance σ_i^2,
- Achievable rate: $R_i \leq \frac{1}{2}\log\left(1 + \frac{P_i}{\sigma_j^2}\right)$ (independent of self-interference!),

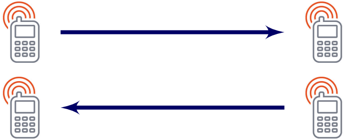


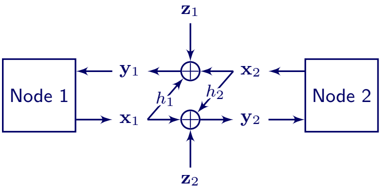


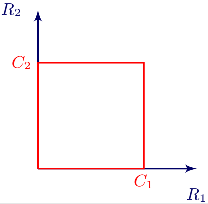


Two-way Channel

- However: Bounds coincide if channel is separable!
- i.e., channel decomposes into two P2P channels.
- The Gaussian channel belongs to this class.
- Inputs: $\mathbf{x}_1$ and $\mathbf{x}_2$ with powers P_1 and P_2,
- Outputs: $\mathbf{y}_i = \mathbf{x}_j + h_i \mathbf{x}_j + \mathbf{z}_i$, $h_i \in \mathbb{R}$, $\mathbf{z}_i$ has variance σ_i^2,
- Achievable rate: $R_i \leq \frac{1}{2} \log \left(1 + \frac{P_i}{\sigma_j^2}\right)$ (independent of self-interference!),
- Capacity [Han 84],





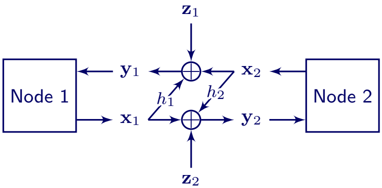


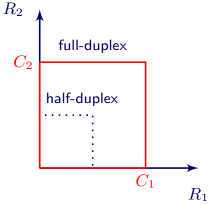
Two-way Channel

Remarks:

Half-duplex vs. full-duplex:

- Half-duplex:
  $R_i \leq \frac{1}{4} \log \left(1 + \frac{P_i}{\sigma_j^2}\right).$
- Full-duplex:
  $R_i \leq \frac{1}{2} \log \left(1 + \frac{P_i}{\sigma_j^2}\right).$
- Full-duplex achieves double rate.



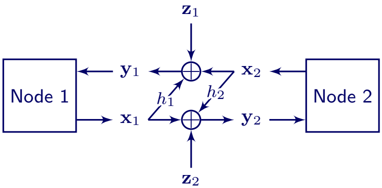


Two-way Channel

Remarks:

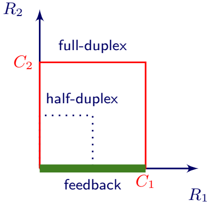
Half-duplex vs. full-duplex:

- Half-duplex:
  $R_i \leq \frac{1}{4}\log\left(1+\frac{P_i}{\sigma_j^2}\right)$.
- Full-duplex:
  $R_i \leq \frac{1}{2}\log\left(1+\frac{P_i}{\sigma_j^2}\right)$.
- Full-duplex achieves double rate.



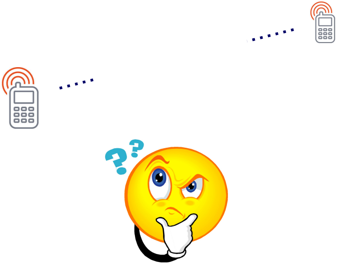
Feedback vs. Two-way:

- Feedback does not increase the P2P capacity [Shannon 56].
- Rate: $R_1 \leq \frac{1}{2}\log\left(1+\frac{P_1}{\sigma_2^2}\right)$.
- Two-way achieves double rate.



Two-way Channel

- What happens if nodes are far/physically separated?



Outline

1. Two-way channel

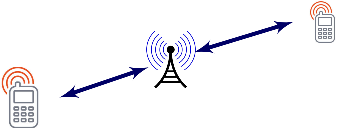
2. Two-way relay channel
   - The linear-deterministic approximation
   - Lattice codes

3. Multi-way relay channel
   - Multi-pair Two-way Relay Channel
   - Multi-way Relay Channel

4. Multi-way Channel

Two-way Relay Channel

Channel with two transceivers and a relay: First studied by [Rankov & Wittneben 06]

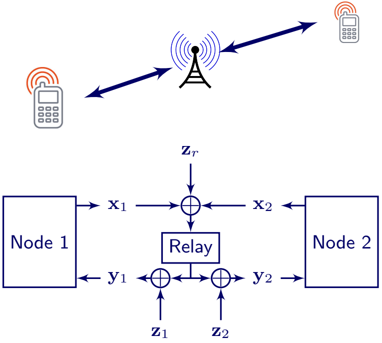


Two-way Relay Channel

Channel with two transceivers and a relay: First studied by [Rankov & Wittneben 06]

Gaussian two-way relay channel:

- Inputs: $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_r$ with powers P_1, P_2, and P_r,

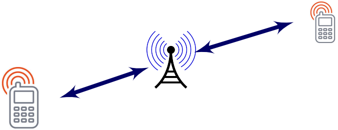


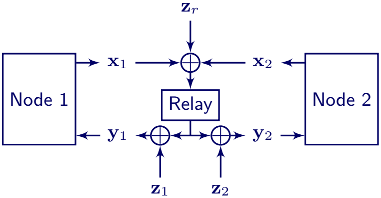
Two-way Relay Channel

Channel with two transceivers and a relay: First studied by [Rankov & Wittneben 06]

Gaussian two-way relay channel:

- Inputs: $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_r$ with powers P_1, P_2, and P_r,
- Outputs: $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$, $\mathbf{y}_i = \mathbf{x}_r + \mathbf{z}_i$, noise variance σ_j^2,



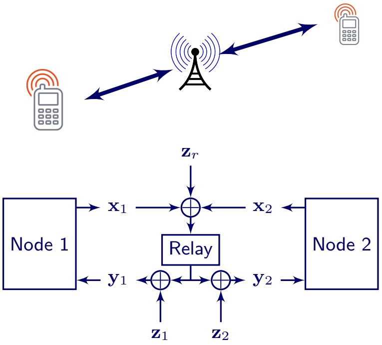


Two-way Relay Channel

Channel with two transceivers and a relay: First studied by [Rankov & Wittneben 06]

Gaussian two-way relay channel:

- Inputs: $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_r$ with powers P_1, P_2, and P_r,
- Outputs: $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$, $\mathbf{y}_i = \mathbf{x}_r + \mathbf{z}_i$, noise variance σ_j^2,
- Goal: Find the capacity region.

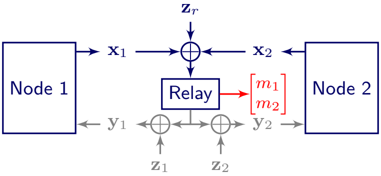


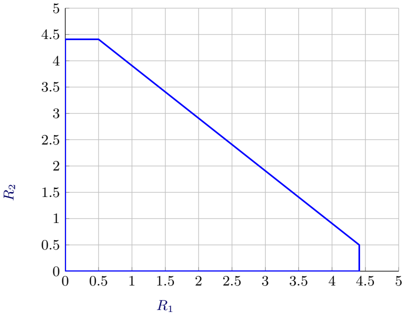
Classical approach

- Treat uplink as a MAC ⇒

$$R_i \leq \frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right)$$

$$R_1 + R_2 \leq \frac{1}{2}\log\left(1+\frac{P_1+P_2}{\sigma^2}\right),$$





Classical approach

- Treat uplink as a MAC $\Rightarrow$

$$R_i \leq \frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right)$$

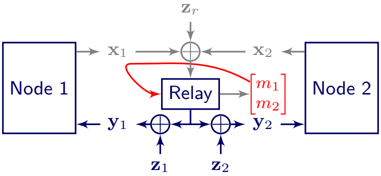
$$R_1+R_2 \leq \frac{1}{2}\log\left(1+\frac{P_1+P_2}{\sigma^2}\right),$$

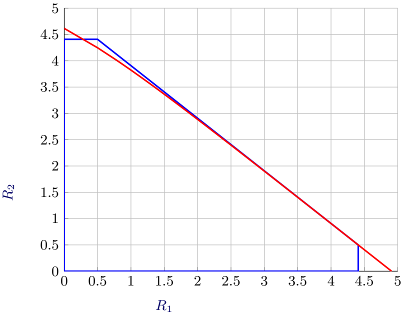
- Treat downlink as a BC $\Rightarrow$

$$R_1 \leq \frac{1}{2}\log\left(1+\frac{p_1}{\sigma_1^2}\right)$$

$$R_2 \leq \frac{1}{2}\log\left(1+\frac{p_2}{\sigma_2^2+p_1}\right)$$

with $p_1+p_2 \leq P_r$,





Classical approach

- Treat uplink as a MAC $\Rightarrow$

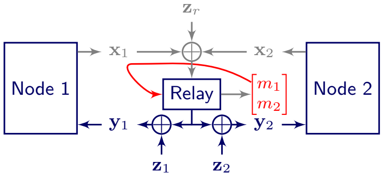
$$R_i \leq \frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right)$$
$$R_1+R_2 \leq \frac{1}{2}\log\left(1+\frac{P_1+P_2}{\sigma^2}\right),$$

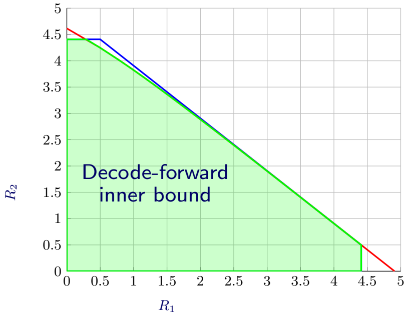
- Treat downlink as a BC $\Rightarrow$

$$R_1 \leq \frac{1}{2}\log\left(1+\frac{p_1}{\sigma_1^2}\right)$$
$$R_2 \leq \frac{1}{2}\log\left(1+\frac{p_2}{\sigma_2^2+p_1}\right)$$

with $p_1+p_2 \leq P_r$,

- Achievable region: intersection





Classical approach

- Treat uplink as a MAC $\Rightarrow$

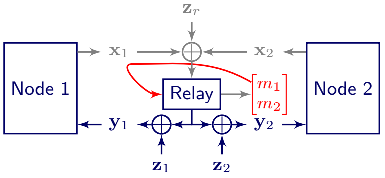
$$R_i \leq \frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right)$$
$$R_1+R_2 \leq \frac{1}{2}\log\left(1+\frac{P_1+P_2}{\sigma^2}\right),$$

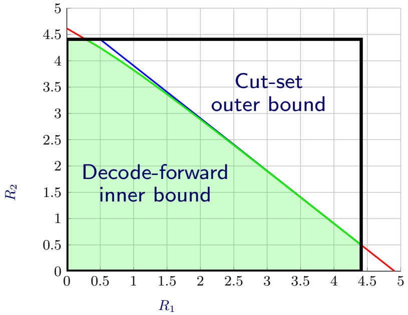
- Treat downlink as a BC $\Rightarrow$

$$R_1 \leq \frac{1}{2}\log\left(1+\frac{p_1}{\sigma_1^2}\right)$$
$$R_2 \leq \frac{1}{2}\log\left(1+\frac{p_2}{\sigma_2^2+p_1}\right)$$

with $p_1+p_2 \leq P_r$,

- Achievable region: intersection
- Cut-set bound:
$R_i \leq$
$\min\left\{\frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right), \frac{1}{2}\log\left(1+\frac{P_r}{\sigma_2^2}\right)\right\}$





Classical approach

- Treat uplink as a MAC $\Rightarrow$

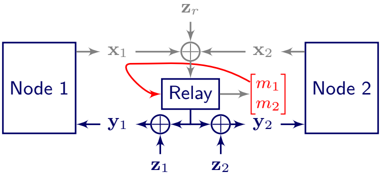
$$R_i \le \frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right)$$
$$R_1+R_2 \le \frac{1}{2}\log\left(1+\frac{P_1+P_2}{\sigma^2}\right),$$

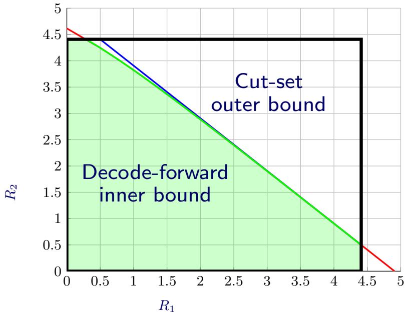
- Treat downlink as a BC $\Rightarrow$

$$R_1 \le \frac{1}{2}\log\left(1+\frac{p_1}{\sigma_1^2}\right)$$
$$R_2 \le \frac{1}{2}\log\left(1+\frac{p_2}{\sigma_2^2+p_1}\right)$$

with $p_1+p_2 \le P_r$,

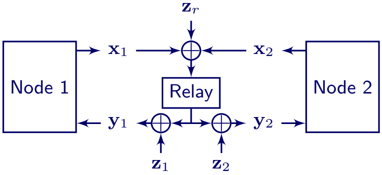
- Achievable region: intersection
- Cut-set bound: $R_i \le \min\left\{\frac{1}{2}\log\left(1+\frac{P_i}{\sigma^2}\right), \frac{1}{2}\log\left(1+\frac{P_r}{\sigma_2^2}\right)\right\}$



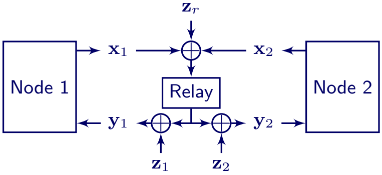


Sum-rate

Let us focus on the sum-rate
$R_\Sigma = R_1 + R_2$:



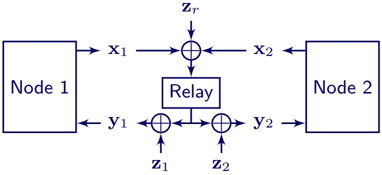
Sum-rate



Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

- simplifying assumption: $P_1 = P_2 = P_r = P$, $\sigma_1^2 = \sigma_2^2 = \sigma^2 = 1$,

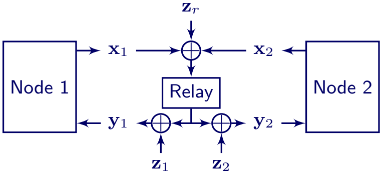
Sum-rate



Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

- simplifying assumption: $P_1 = P_2 = P_r = P$, $\sigma_1^2 = \sigma_2^2 = \sigma^2 = 1$,
- MAC: $R_\Sigma \leq \frac{1}{2}\log(1 + 2P)$,

Sum-rate



Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

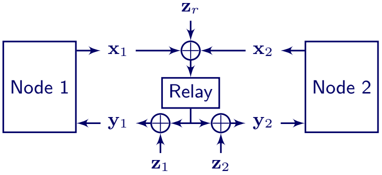
- simplifying assumption: $P_1 = P_2 = P_r = P$, $\sigma_1^2 = \sigma_2^2 = \sigma^2 = 1$,
- MAC: $R_\Sigma \leq \frac{1}{2}\log(1 + 2P)$,
- BC: $R_\Sigma \leq \frac{1}{2}\log(1 + P)$,

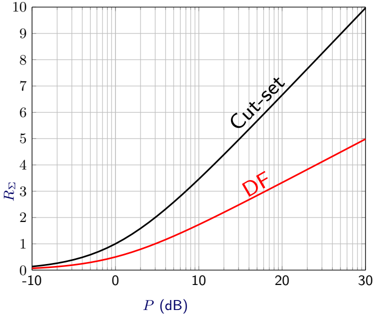
Sum-rate

Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

- simplifying assumption: $P_1 = P_2 = P_r = P$, $\sigma_1^2 = \sigma_2^2 = \sigma^2 = 1$,
- MAC: $R_\Sigma \leq \frac{1}{2}\log(1+2P)$,
- BC: $R_\Sigma \leq \frac{1}{2}\log(1+P)$,

$\Rightarrow$ max $R_{DF} = \frac{1}{2}\log(1+P)$





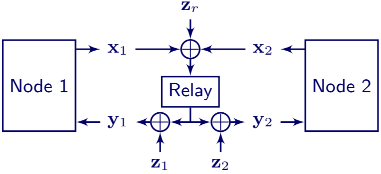
Sum-rate

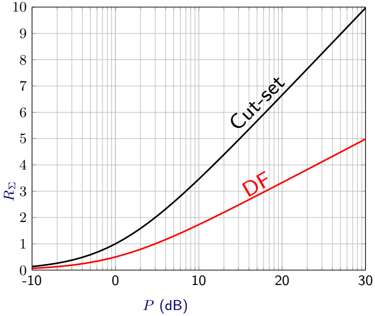
Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

- simplifying assumption:
  $P_1 = P_2 = P_r = P$,
  $\sigma_1^2 = \sigma_2^2 = \sigma^2 = 1$,
- MAC: $R_\Sigma \le \frac{1}{2}\log(1+2P)$,
- BC: $R_\Sigma \le \frac{1}{2}\log(1+P)$,

$\Rightarrow$ max $R_{DF} = \frac{1}{2}\log(1+P)$

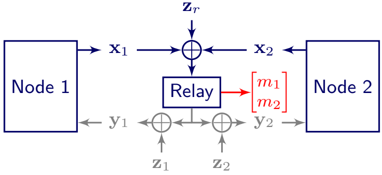
- Question: How to improve?





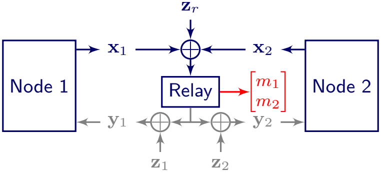
Network Coding

- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.



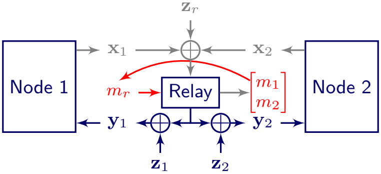
Network Coding

- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.



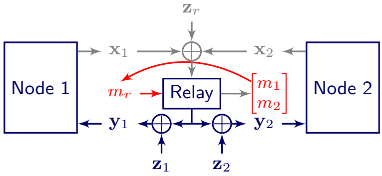
Network Coding

- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,



Network Coding

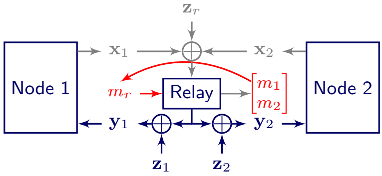
- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,
- Node i decodes $\mathbf{x}_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,



Network Coding

- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,
- Node i decodes $\mathbf{x}_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,

$\Rightarrow$ $R_r \leq \frac{1}{2}\log(1+P) \Rightarrow$ $R_\Sigma \leq \log(1+P)$

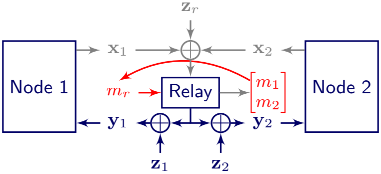


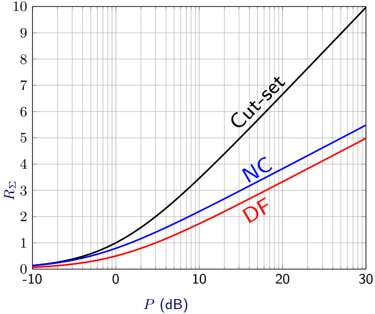
Network Coding

- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,
- Node i decodes $\mathbf{x}_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,

$\Rightarrow$ $R_r \leq \frac{1}{2}\log(1+P) \Rightarrow$ $R_\Sigma \leq \log(1+P)$

$\Rightarrow$ $R_{NC} = \frac{1}{2}\log(1+2P)$ (alternative: BC with side info.)





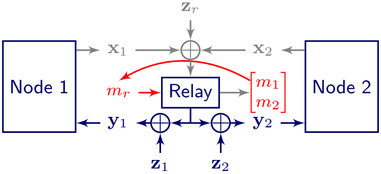
Network Coding

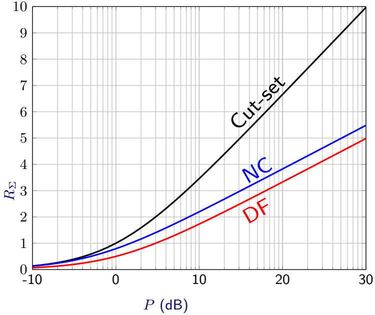
- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,
- Node i decodes $\mathbf{x}_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,

$\Rightarrow$ $R_r \leq \frac{1}{2}\log(1+P) \Rightarrow$ $R_\Sigma \leq \log(1+P)$

$\Rightarrow$ $R_{NC} = \frac{1}{2}\log(1+2P)$ (alternative: BC with side info.)

- Good at low SNR, but not at high SNR





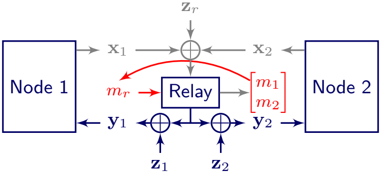
Network Coding

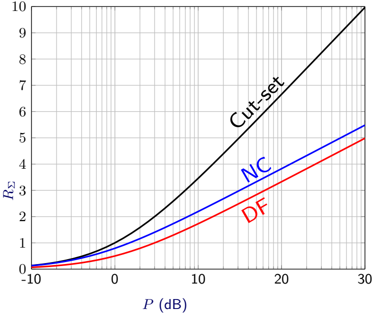
- Treat uplink as a MAC $\Rightarrow$ $R_\Sigma \leq \frac{1}{2}\log(1+2P)$.
- Relay obtains $\mathbf{x}_1(m_1)$ and $\mathbf{x}_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $\mathbf{x}_r(m_r)$,
- Node i decodes $\mathbf{x}_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,

$\Rightarrow$ $R_r \leq \frac{1}{2}\log(1+P) \Rightarrow$ $R_\Sigma \leq \log(1+P)$

$\Rightarrow$ $R_{NC} = \frac{1}{2}\log(1+2P)$ (alternative: BC with side info.)

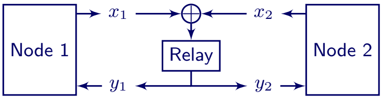
- Good at low SNR, but not at high SNR
- Further improvement?





Improvement

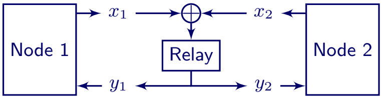
Important insight: Relay does not need to decode the messages!



Improvement

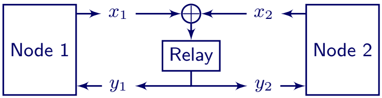
Important insight: Relay does not need to decode the messages!

Example: Binary additive noiseless channel



Improvement

Important insight: Relay does not need to decode the messages!

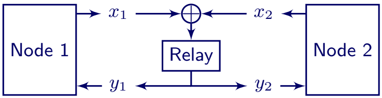


Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,

Improvement

Important insight: Relay does not need to decode the messages!

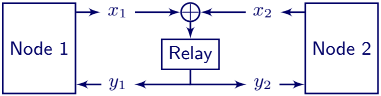


Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,

Improvement

Important insight: Relay does not need to decode the messages!

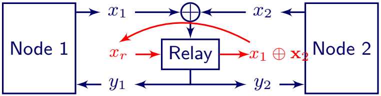


Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,
- NC: Relay decodes x_1 then x_2 (two transmissions), and constructs $x_r = x_1 \oplus x_2$

Improvement

Important insight: Relay does not need to decode the messages!

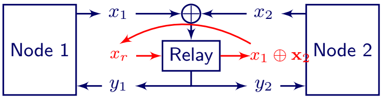


Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,
- NC: Relay decodes x_1 then x_2 (two transmissions), and constructs $x_r = x_1 \oplus x_2$
- Additive binary-noiseless channel: Relay can decode $x_r = x_1 \oplus x_2$ directly (one transmission!),

Improvement

Important insight: Relay does not need to decode the messages!



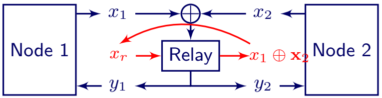
Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,
- NC: Relay decodes x_1 then x_2 (two transmissions), and constructs $x_r = x_1 \oplus x_2$
- Additive binary-noiseless channel: Relay can decode $x_r = x_1 \oplus x_2$ directly (one transmission!),
- But: Physical channels are not binary and not noiseless!

Improvement

Important insight: Relay does not need to decode the messages!

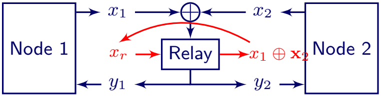
Example: Binary additive noiseless channel



- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,
- NC: Relay decodes x_1 then x_2 (two transmissions), and constructs $x_r = x_1 \oplus x_2$
- Additive binary-noiseless channel: Relay can decode $x_r = x_1 \oplus x_2$ directly (one transmission!),
- But: Physical channels are not binary and not noiseless!
- However: A binary-noiseless approximation exists!

Improvement

Important insight: Relay does not need to decode the messages!



Example: Binary additive noiseless channel

- Inputs: $x_1, x_2, x_r \in \mathbb{F}_2$,
- Outputs: $y_r = x_1 \oplus x_2$, $y_i = x_r$,
- NC: Relay decodes x_1 then x_2 (two transmissions), and constructs $x_r = x_1 \oplus x_2$
- Additive binary-noiseless channel: Relay can decode $x_r = x_1 \oplus x_2$ directly (one transmission!),
- But: Physical channels are not binary and not noiseless!
- However: A binary-noiseless approximation exists!

[Avestimehr *et al.* 07]

Analyse using the Linear Deterministic (LD) Model

Outline

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,
- Write the binary representation

$$y = 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i}$$

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,
- Write the binary representation

$$y = 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i}$$

$$\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \quad \text{assuming } |x| \leq 1$$

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,
- Write the binary representation

$$y = 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i}$$

$$\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \quad \text{assuming } |x| \leq 1$$

$$\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \quad \text{assuming } |z| < 1$$

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,
- Write the binary representation

$$
\begin{aligned}
y &= 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \\
&\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \quad \text{assuming } |x| \leq 1 \\
&\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \quad \text{assuming } |z| < 1 \\
&\approx 2^{n} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \quad \text{defining } n = \left\lceil \frac{1}{2}\log(P) \right\rceil
\end{aligned}
$$

The linear-deterministic approximation

- Let x and z have unit power and $y = \sqrt{P}x + z$,
- Write the binary representation

$$
\begin{aligned}
y &= 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \\
&\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=-\infty}^{\infty} z_i 2^{-i} \quad \text{assuming } |x| \leq 1 \\
&\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \quad \text{assuming } |z| < 1 \\
&\approx 2^{n} \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \quad \text{defining } n = \left\lceil \frac{1}{2}\log(P) \right\rceil \\
&= 2^{n} \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \quad \text{regrouping}
\end{aligned}
$$

The linear-deterministic approximation

$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

The linear-deterministic approximation

$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

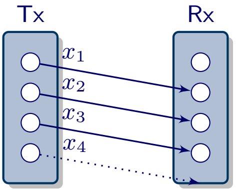
- Ignore the noisy bits $\Rightarrow$ $y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}$

The linear-deterministic approximation

$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

- Ignore the noisy bits $\Rightarrow$ $y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}$

$\Rightarrow$ Channel can be modelled as n bit-pipes!



Impact of noise modelled by clipping the least-significant bits

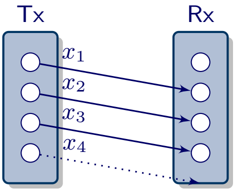
The linear-deterministic approximation

$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

- Ignore the noisy bits ⇒ $y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}$

⇒ Channel can be modelled as n bit-pipes!

⇒ $C \approx n = \left\lceil \frac{1}{2} \log(P) \right\rceil$



Impact of noise modelled by clipping the least-significant bits

The linear-deterministic approximation

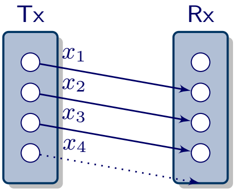
$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

- Ignore the noisy bits $\Rightarrow$ $y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}$

$\Rightarrow$ Channel can be modelled as n bit-pipes!

$\Rightarrow$ $C \approx n = \left\lceil \frac{1}{2} \log(P) \right\rceil$

- A good approximation at high SNR



Impact of noise modelled by clipping the least-significant bits

The linear-deterministic approximation

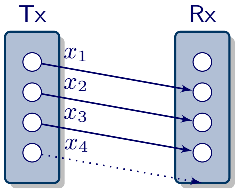
$$y \approx \underbrace{2^n \sum_{i=1}^{n} x_i 2^{-i}}_{\text{noiseless bits}} + \underbrace{\sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}}_{\text{noisy bits}},$$

- Ignore the noisy bits $\Rightarrow$ $y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}$

$\Rightarrow$ Channel can be modelled as n bit-pipes!

$\Rightarrow$ $C \approx n = \left\lceil \frac{1}{2} \log(P) \right\rceil$

- A good approximation at high SNR



Impact of noise modelled by clipping the least-significant bits

Deterministic P2P

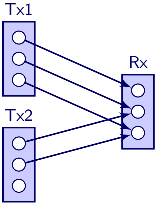
A Gaussian P2P can be approximated as a binary channel with input $\mathbf{x} = [x_1,\ x_2, \cdots, x_q]^T$ and output $\mathbf{y} = \mathbf{S}^{q-n}\mathbf{x}$ where $q \geq n = \left\lceil \frac{1}{2} \log(P) \right\rceil$ and $\mathbf{S} = \begin{bmatrix} \mathbf{0} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$ is a down-ward shift matrix.

The linear-deterministic approximation

Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i) \right\rceil$ and
  $q = \max\{n_1, n_2\}$,

MAC with $n_1 \geq n_2$

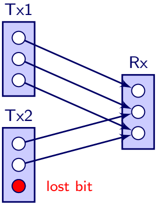


The linear-deterministic approximation

Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i)\right\rceil$ and
  $q = \max\{n_1, n_2\}$,

MAC with $n_1 \geq n_2$

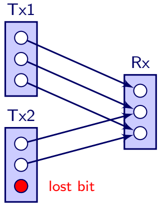


The linear-deterministic approximation

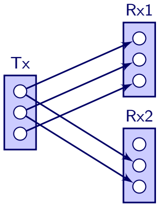
Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i) \right\rceil$ and $q = \max\{n_1, n_2\}$,
- BC: $\mathbf{y}_i = \mathbf{S}^{q-n_i}\mathbf{x}$,

MAC with $n_1 \geq n_2$



BC with $n_1 \geq n_2$

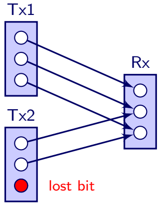


The linear-deterministic approximation

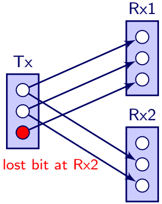
Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i)\right\rceil$ and
  $q = \max\{n_1, n_2\}$,
- BC: $\mathbf{y}_i = \mathbf{S}^{q-n_i}\mathbf{x}$,

MAC with $n_1 \geq n_2$



BC with $n_1 \geq n_2$

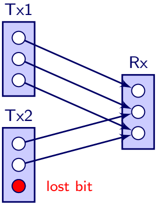


The linear-deterministic approximation

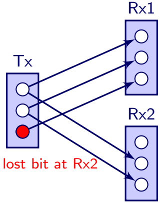
Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i) \right\rceil$ and $q = \max\{n_1, n_2\}$,
- BC: $\mathbf{y}_i = \mathbf{S}^{q-n_i}\mathbf{x}$,
- A very useful tool for studying Gaussian networks

MAC with $n_1 \geq n_2$



BC with $n_1 \geq n_2$

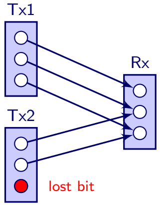


The linear-deterministic approximation

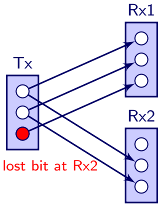
Similar approximation can be applied to the:

- MAC:
  $\mathbf{y} = \mathbf{S}^{q-n_1}\mathbf{x}_1 \oplus \mathbf{S}^{q-n_2}\mathbf{x}_2$
  where $n_i = \left\lceil \frac{1}{2}\log(P_i) \right\rceil$ and $q = \max\{n_1, n_2\}$,
- BC: $\mathbf{y}_i = \mathbf{S}^{q-n_i}\mathbf{x}$,
- A very useful tool for studying Gaussian networks
- Obtained insights in an LD network can be extended to corresponding Gaussian networks

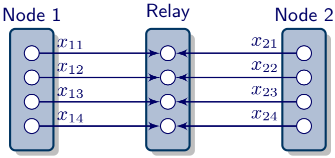
MAC with $n_1 \geq n_2$



BC with $n_1 \geq n_2$

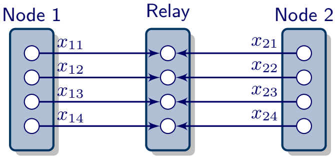


LD Two-way relay channel

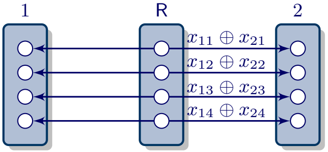


- Uplink: MAC with $n_1 = n_2 = n = 4$,
- Node i sends $\mathbf{x}_i \in \mathbf{F}_2^q$,
- Relay receives $\mathbf{S}^{q-n}\mathbf{x}_1 \oplus \mathbf{S}^{q-n}\mathbf{x}_2$,
- $\mathbf{x}_1 \oplus \mathbf{x}_2$ decodable if $\max\{R_1, R_2\} \leq n$ (send on the most-significant bits)

LD Two-way relay channel

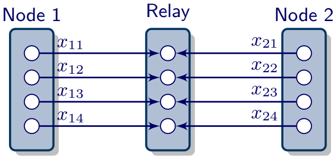


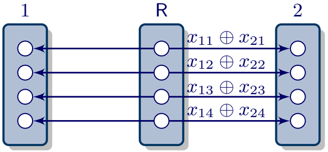
- Uplink: MAC with $n_1 = n_2 = n = 4$,
- Node i sends $\mathbf{x}_i \in \mathbf{F}_2^q$,
- Relay receives $\mathbf{S}^{q-n}\mathbf{x}_1 \oplus \mathbf{S}^{q-n}\mathbf{x}_2$,
- $\mathbf{x}_1 \oplus \mathbf{x}_2$ decodable if $\max\{R_1, R_2\} \leq n$ (send on the most-significant bits)



- Downlink: BC with $n_1 = n_2 = n = 4$,
- Relay sends $\mathbf{x}_r = \mathbf{x}_1 \oplus \mathbf{x}_2$ with rate $R_r = \max\{R_1, R_2\}$,
- Node i receives $\mathbf{S}^{q-n}\mathbf{x}_r$,
- $\mathbf{x}_r$ decodable if $R_r \leq n$ (send on the most-significant bits),

LD Two-way relay channel





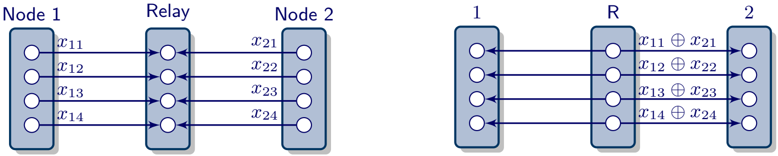
- Uplink: MAC with $n_1 = n_2 = n = 4$,
- Node i sends $\mathbf{x}_i \in \mathbf{F}_2^q$,
- Relay receives $\mathbf{S}^{q-n}\mathbf{x}_1 \oplus \mathbf{S}^{q-n}\mathbf{x}_2$,
- $\mathbf{x}_1 \oplus \mathbf{x}_2$ decodable if $\max\{R_1, R_2\} \leq n$ (send on the most-significant bits)

- Downlink: BC with $n_1 = n_2 = n = 4$,
- Relay sends $\mathbf{x}_r = \mathbf{x}_1 \oplus \mathbf{x}_2$ with rate $R_r = \max\{R_1, R_2\}$,
- Node i receives $\mathbf{S}^{q-n}\mathbf{x}_r$,
- $\mathbf{x}_r$ decodable if $R_r \leq n$ (send on the most-significant bits),

Compute-forward (a.k.a. Physical-layer NC)

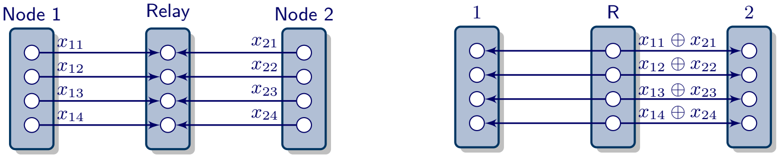
Relay decodes a function (sum) of the transmit signals, and forwards this sum. Each node can decode the desired signal using its own signal as side information.

LD Two-way relay channel



- Achievable rate $\max\{R_1, R_2\} \leq n = \left\lceil \frac{1}{2} \log(P) \right\rceil$.

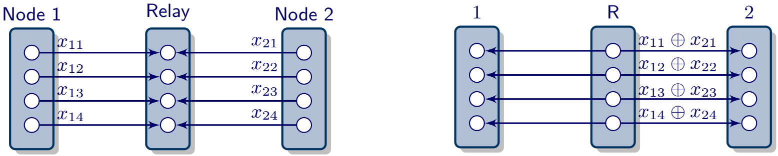
LD Two-way relay channel



- Achievable rate $\max\{R_1, R_2\} \leq n = \left\lceil \frac{1}{2} \log(P) \right\rceil$.

$\Rightarrow$ $R_{CF} = 2n = 2\left\lceil \frac{1}{2} \log(P) \right\rceil \approx \log(P)$

LD Two-way relay channel

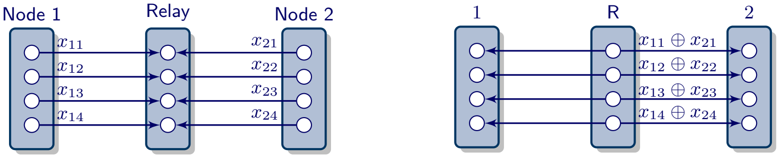


- Achievable rate $\max\{R_1, R_2\} \leq n = \left\lceil \frac{1}{2}\log(P) \right\rceil$.

$\Rightarrow$ $R_{CF} = 2n = 2\left\lceil \frac{1}{2}\log(P) \right\rceil \approx \log(P)$

- vs. $R_{NC} = \frac{1}{2}\log(1+2P)$ for network coding!

LD Two-way relay channel



- Achievable rate $\max\{R_1, R_2\} \leq n = \left\lceil \frac{1}{2} \log(P) \right\rceil$.

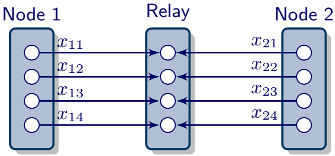
$\Rightarrow$ $R_{CF} = 2n = 2\left\lceil \frac{1}{2} \log(P) \right\rceil \approx \log(P)$

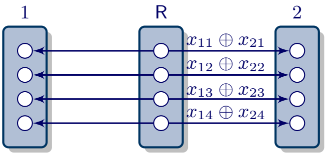
- vs. $R_{NC} = \frac{1}{2} \log(1 + 2P)$ for network coding!

Compute-forward (CF)

CF (almost) doubles the rate in comparison to DF and to NC (at high SNR).

LD Two-way relay channel





- Achievable rate $\max\{R_1, R_2\} \leq n = \left\lceil \frac{1}{2} \log(P) \right\rceil$.

$\Rightarrow$ $R_{CF} = 2n = 2\left\lceil \frac{1}{2} \log(P) \right\rceil \approx \log(P)$

- vs. $R_{NC} = \frac{1}{2} \log(1 + 2P)$ for network coding!

Compute-forward (CF)

CF (almost) doubles the rate in comparison to DF and to NC (at high SNR).

How to extend to Gaussian two-way relay channels?

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.
- Important requirement: Sum of two codewords is a codeword.

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.
- Important requirement: Sum of two codewords is a codeword.
- True in the LD channel: XOR of two bits is a bit.

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.
- Important requirement: Sum of two codewords is a codeword.
- True in the LD channel: XOR of two bits is a bit.
- Gaussian channels: normally random codes!

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.
- Important requirement: Sum of two codewords is a codeword.
- True in the LD channel: XOR of two bits is a bit.
- Gaussian channels: normally random codes!
- sum of two random codewords is not necessarily a codeword!

LD Two-way relay channel

- In CF, relay decodes the sum of input codewords.
- Important requirement: Sum of two codewords is a codeword.
- True in the LD channel: XOR of two bits is a bit.
- Gaussian channels: normally random codes!
- sum of two random codewords is not necessarily a codeword!
- Now what?

Outline

1. Two-way channel

2. Two-way relay channel
   - The linear-deterministic approximation
   - Lattice codes

3. Multi-way relay channel
   - Multi-pair Two-way Relay Channel
   - Multi-way Relay Channel

4. Multi-way Channel

Computation

Definition (Computation)

Computation is the process of recovering a function of transmit codewords from a received sequence of symbols after sending the codewords through a channel.

How?

Computation

Definition (Computation)

Computation is the process of recovering a function of transmit codewords from a received sequence of symbols after sending the codewords through a channel.

How?

Computation can be accomplished by using lattice codes.

Idea: Codes located on a grid so that the sum of two codewords is a codeword.

Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

Examples:

- $\mathbb{Z}$ is a one-dimensional lattice

Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

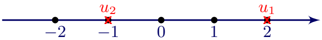
Examples:

- $\mathbb{Z}$ is a one-dimensional lattice

Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

Examples:

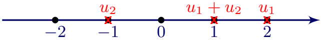


- $\mathbb{Z}$ is a one-dimensional lattice
- two codewords: $u_1, u_2 \in \mathbb{Z}$,

Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

Examples:

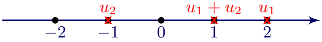


- $\mathbb{Z}$ is a one-dimensional lattice
- two codewords: $u_1, u_2 \in \mathbb{Z}$,
- $u_1 + u_2 \in \mathbb{Z}$ also a codeword,

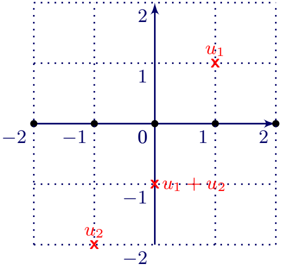
Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow$ $u_1 + u_2$ lattice code!

Examples:



- $\mathbb{Z}$ is a one-dimensional lattice
- two codewords: $u_1, u_2 \in \mathbb{Z}$,
- $u_1 + u_2 \in \mathbb{Z}$ also a codeword,
- $\mathbb{Z}^2$ is a 2-dimensional lattice,

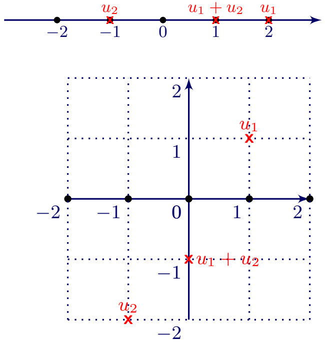


Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

Examples:

- $\mathbb{Z}$ is a one-dimensional lattice
- two codewords: $u_1, u_2 \in \mathbb{Z}$,
- $u_1 + u_2 \in \mathbb{Z}$ also a codeword,
- $\mathbb{Z}^2$ is a 2-dimensional lattice,



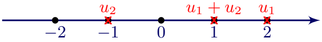
Nested-lattice codes

Nested lattice codes achieve the capacity $\frac{1}{2}\log(1+P)$ of the P2P channel.

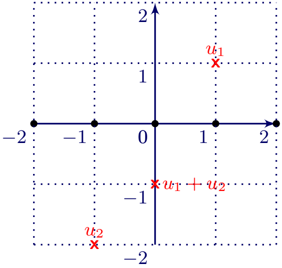
Lattice-codes

Property: u_1 and u_2 lattice codes $\Rightarrow u_1 + u_2$ lattice code!

Examples:



- $\mathbb{Z}$ is a one-dimensional lattice
- two codewords: $u_1, u_2 \in \mathbb{Z}$,
- $u_1 + u_2 \in \mathbb{Z}$ also a codeword,
- $\mathbb{Z}^2$ is a 2-dimensional lattice,



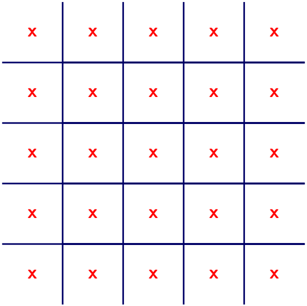
Nested-lattice codes

Nested lattice codes achieve the capacity $\frac{1}{2}\log(1+P)$ of the P2P channel.

What is a nested lattice code?

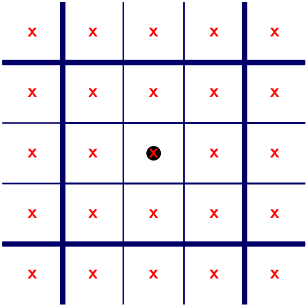
Nested-lattice codes

- **x**: Fine lattice Λ_f



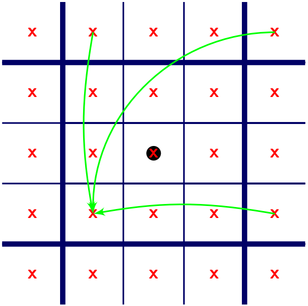
Nested-lattice codes

- **x**: Fine lattice Λ_f
- Coarse lattice $\Lambda_c \subset \Lambda_f$



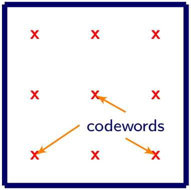
Nested-lattice codes

- **x**: Fine lattice Λ_f
- Coarse lattice $\Lambda_c \subset \Lambda_f$
- Modulo Λ_c



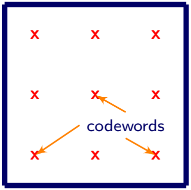
Nested-lattice codes

- **x**: Fine lattice Λ_f
- Coarse lattice $\Lambda_c \subset \Lambda_f$
- Modulo Λ_c
- Nested lattice code



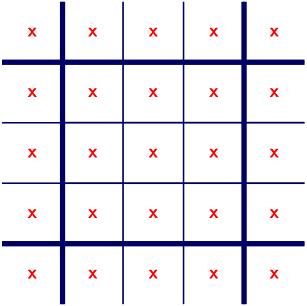
Nested-lattice codes

- **x**: Fine lattice Λ_f
- Coarse lattice $\Lambda_c \subset \Lambda_f$
- Modulo Λ_c
- Nested lattice code
- Power Constraint satisfied by the choice of Λ_c



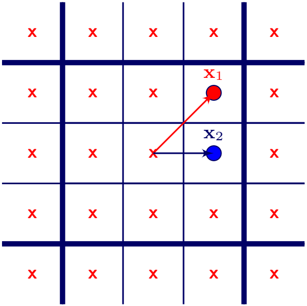
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,



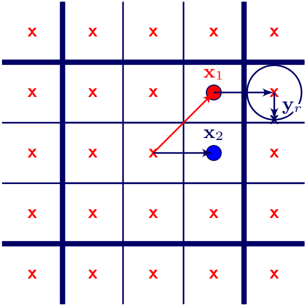
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,



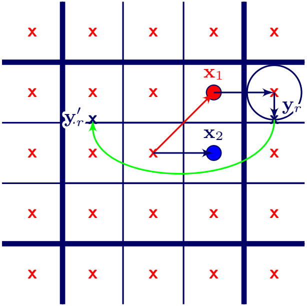
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,



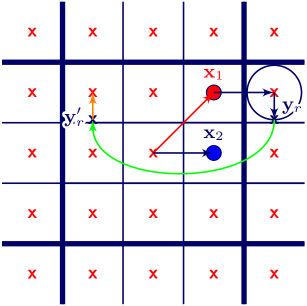
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,
- calculates $\mathbf{y}'_r = \mathbf{y}_r \bmod \Lambda_c$,



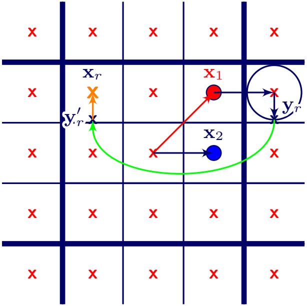
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,
- calculates $\mathbf{y}'_r = \mathbf{y}_r \bmod \Lambda_c$,
- decode to the nearest codeword,



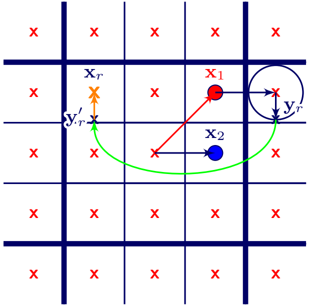
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,
- calculates $\mathbf{y}_r' = \mathbf{y}_r \bmod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $\mathbf{x}_r = (\mathbf{x}_1 + \mathbf{x}_2) \bmod \Lambda_c$



Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,
- calculates $\mathbf{y}'_r = \mathbf{y}_r \bmod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $\mathbf{x}_r = (\mathbf{x}_1 + \mathbf{x}_2) \bmod \Lambda_c$



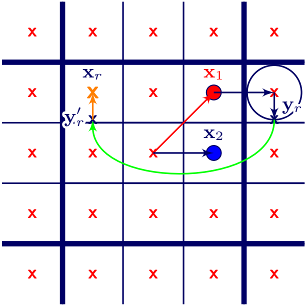
Remarks:

- $\mathbf{x}_r$ belongs to the same nested lattice codebook

$\Rightarrow$ $R_r = R$ (same rate as $\mathbf{x}_1$ and $\mathbf{x}_2$)

Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power P,
- send $\mathbf{x}_1$ and $\mathbf{x}_2$,
- Relay receives $\mathbf{y}_r = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}_r$,
- calculates $\mathbf{y}'_r = \mathbf{y}_r \bmod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $\mathbf{x}_r = (\mathbf{x}_1 + \mathbf{x}_2) \bmod \Lambda_c$



Remarks:

- $\mathbf{x}_r$ belongs to the same nested lattice codebook

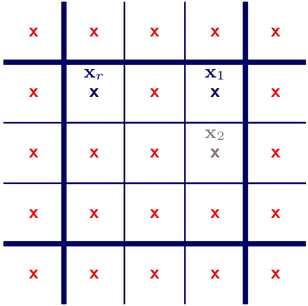
$\Rightarrow$ $R_r = R$ (same rate as $\mathbf{x}_1$ and $\mathbf{x}_2$)

Computation rate [Nazer & Gastpar 11]

Relay can compute $(\mathbf{x}_1 + \mathbf{x}_2) \bmod \Lambda_c$ as long as $R \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + P\right)\right]^+$.

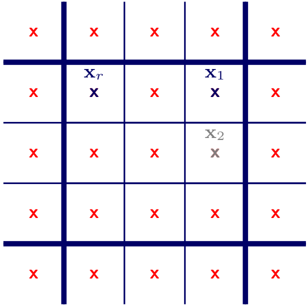
Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),



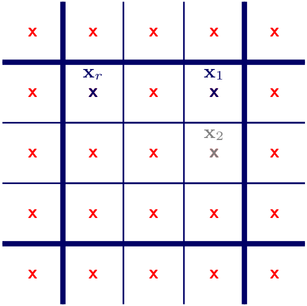
Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),
- node i decodes $\mathbf{x}_r \Rightarrow$ $R \leq \frac{1}{2}\log(1+P)$.



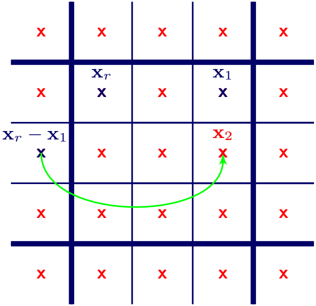
Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),
- node i decodes $\mathbf{x}_r \Rightarrow$ $R \leq \frac{1}{2}\log(1+P)$.
- Can $\mathbf{x}_j$ be recovered from $\mathbf{x}_r$?



Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),
- node i decodes $\mathbf{x}_r \Rightarrow$ $R \leq \frac{1}{2}\log(1+P)$.
- Can $\mathbf{x}_j$ be recovered from $\mathbf{x}_r$?
- Yes! Node i calculates $(\mathbf{x}_r - \mathbf{x}_i) \bmod \Lambda_c$,

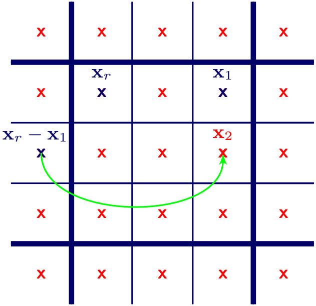


Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),
- node i decodes $\mathbf{x}_r \Rightarrow$ $R \leq \frac{1}{2}\log(1+P)$.
- Can $\mathbf{x}_j$ be recovered from $\mathbf{x}_r$?
- Yes! Node i calculates $(\mathbf{x}_r - \mathbf{x}_i) \bmod \Lambda_c$,
- CF rate constraints $R = \max\{R_1, R_2\}$

$$R \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+ \quad \text{uplink}$$

$$R \leq \frac{1}{2}\log\left(1+P\right) \quad \text{downlink}$$

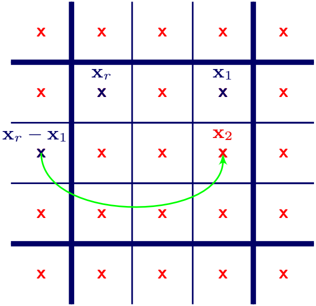


Back to the two-way relay channel

- Relay sends $\mathbf{x}_r$ (rate R),
- node i decodes $\mathbf{x}_r \Rightarrow$ $R \leq \frac{1}{2}\log(1+P)$.
- Can $\mathbf{x}_j$ be recovered from $\mathbf{x}_r$?
- Yes! Node i calculates $(\mathbf{x}_r - \mathbf{x}_i) \bmod \Lambda_c$,
- CF rate constraints $R = \max\{R_1, R_2\}$

$$R \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+ \quad \text{uplink}$$

$$R \leq \frac{1}{2}\log\left(1+P\right) \quad \text{downlink}$$



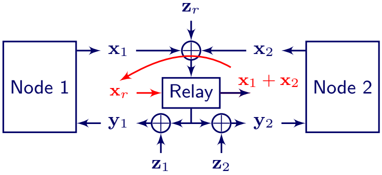
Max sum-rate for CF: $R_{CF} = \left[\log\left(\frac{1}{2}+P\right)\right]^+$

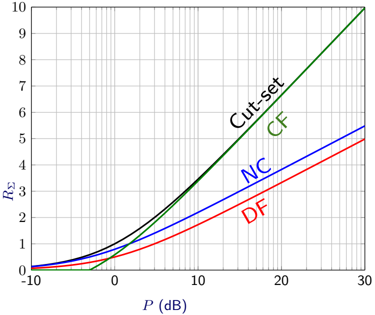
CF sum-rate

$$R_{DF} = \frac{1}{2}\log(1+P)$$

$$R_{NC} = \frac{1}{2}\log(1+2P)$$

$$R_{CF} = \left[\log\left(\frac{1}{2}+P\right)\right]^{+}.$$

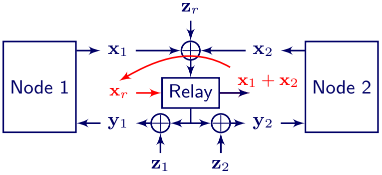


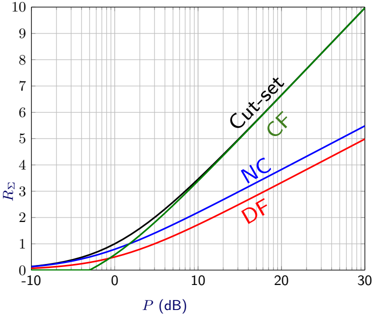


CF sum-rate

$$R_{DF} = \frac{1}{2}\log(1+P)$$
$$R_{NC} = \frac{1}{2}\log(1+2P)$$
$$R_{CF} = \left[\log\left(\frac{1}{2}+P\right)\right]^{+}.$$

- CF doubles the rate (at high SNR),





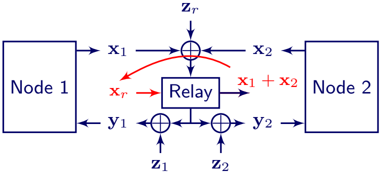
CF sum-rate

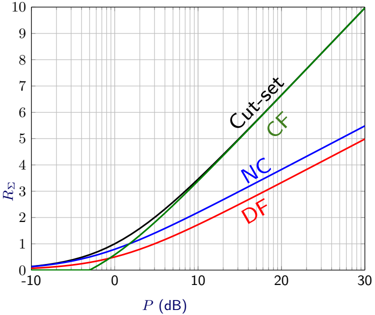
$$R_{DF} = \frac{1}{2}\log(1+P)$$
$$R_{NC} = \frac{1}{2}\log(1+2P)$$
$$R_{CF} = \left[\log\left(\frac{1}{2}+P\right)\right]^+.$$

- CF doubles the rate (at high SNR),

+ Close to capacity at high SNR

- zero rate at low SNR

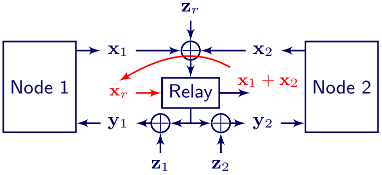


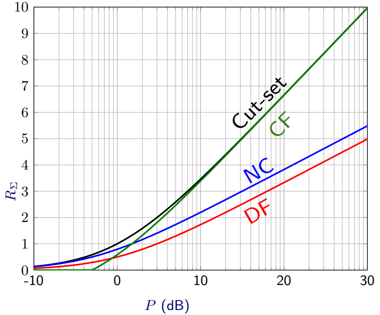


CF sum-rate

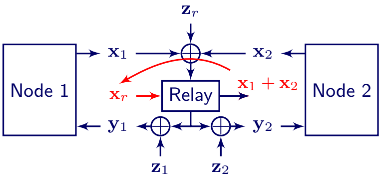
$$R_{DF} = \frac{1}{2}\log(1+P)$$
$$R_{NC} = \frac{1}{2}\log(1+2P)$$
$$R_{CF} = \left[\log\left(\frac{1}{2}+P\right)\right]^+.$$

- CF doubles the rate (at high SNR),
- \+ Close to capacity at high SNR
- \- zero rate at low SNR
- Best scheme: combination of CF and NC.



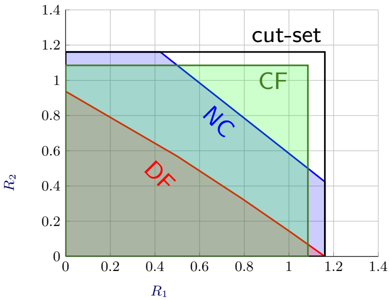


CF rate region



- CF achieves

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+.$$

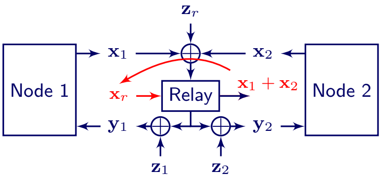


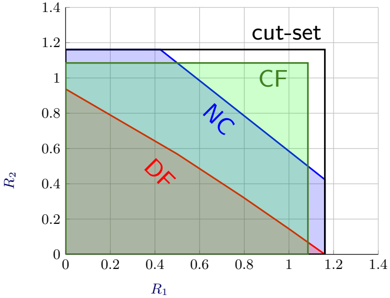
CF rate region

- CF achieves

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+.$$

- highest sum-rate



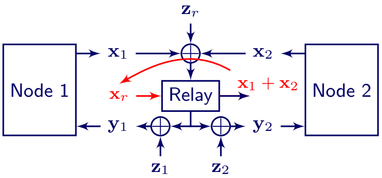


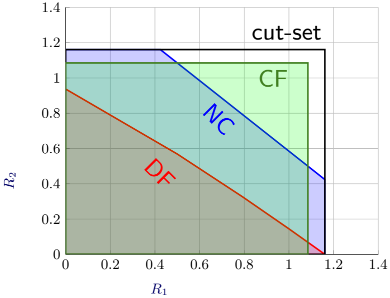
CF rate region

- CF achieves

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+.$$

- highest sum-rate
- But: NC is better in some regions,



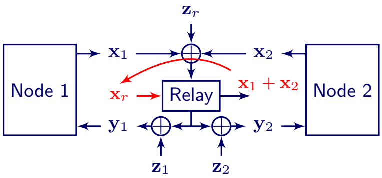


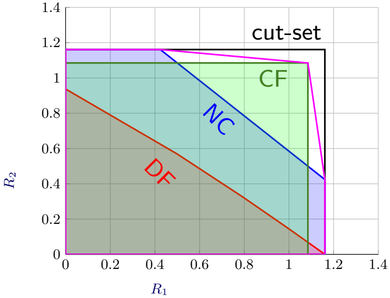
CF rate region

- CF achieves

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+.$$

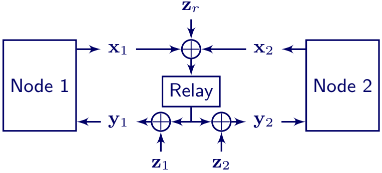
- highest sum-rate
- But: NC is better in some regions,
- Best: Time-sharing NC and CF,





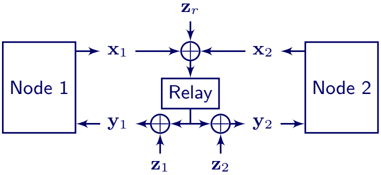
CF rate region

What happens if $P_1 \geq P_2$ and P_r arbitrary?



CF rate region

What happens if $P_1 \geq P_2$ and P_r arbitrary?



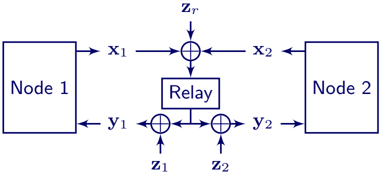
- Reduce P_1 to P_2 and use CF:

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + P_2\right)\right]^+$$

$$R_1, R_2 \leq \frac{1}{2}\log\left(1 + P_r\right)$$

CF rate region

What happens if $P_1 \geq P_2$ and P_r arbitrary?

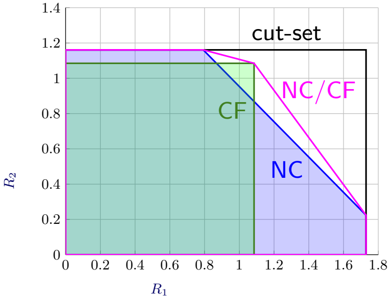


- Reduce P_1 to P_2 and use CF:

$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + P_2\right)\right]^+$$

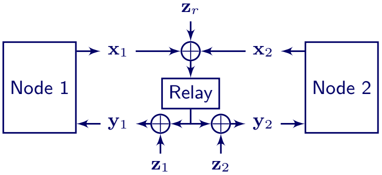
$$R_1, R_2 \leq \frac{1}{2}\log\left(1 + P_r\right)$$

- Time-sharing NC and CF,



CF rate region

What happens if $P_1 \geq P_2$ and P_r arbitrary?

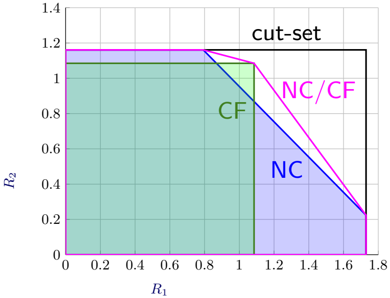


- Reduce P_1 to P_2 and use CF:

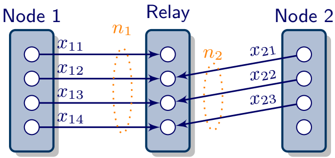
$$R_1, R_2 \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + P_2\right)\right]^+$$

$$R_1, R_2 \leq \frac{1}{2}\log(1 + P_r)$$

- Time-sharing NC and CF,
- Far from sum-capacity!
- Can we do better?

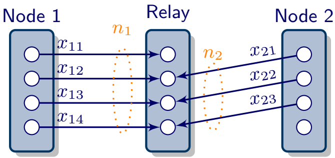


Back to LD

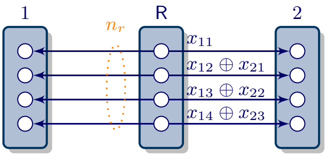


- $P_1 \geq P_2 \Rightarrow n_1 \geq n_2$,
- use n_2 bits for CF,
- use $n_1 - n_2$ bits for DF,
- R_1 and R_2 achievable if $R_1 \leq n_1$ and $R_2 \leq n_2$

Back to LD

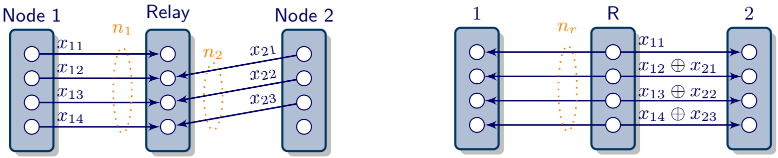


- $P_1 \geq P_2 \Rightarrow n_1 \geq n_2$,
- use n_2 bits for CF,
- use $n_1 - n_2$ bits for DF,
- R_1 and R_2 achievable if $R_1 \leq n_1$ and $R_2 \leq n_2$



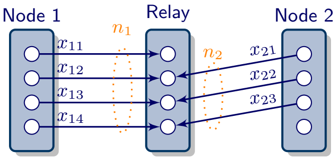
- n_r bit-pipes,
- node 2 gets $\mathbf{x}_1$,
- node 1 gets $\mathbf{x}_2$,
- R_1 and R_2 achievable if $\max\{R_1, R_2\} \leq n_r$,

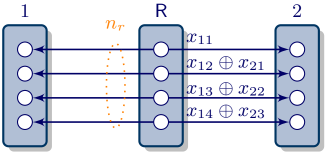
LD Two-way relay channel



- Achievable rates: $R_1 \leq \min\{n_1, n_r\}$ and $R_2 \leq \min\{n_2, n_r\}$.

LD Two-way relay channel

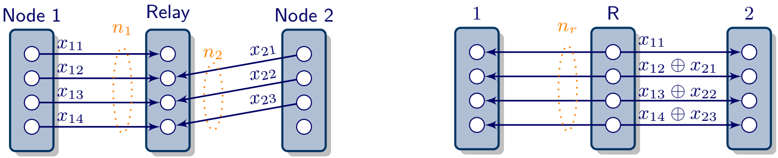




- Achievable rates: $R_1 \leq \min\{n_1, n_r\}$ and $R_2 \leq \min\{n_2, n_r\}$.

$\Rightarrow$ $R_1 \leq \lceil \frac{1}{2} \log(\min\{P_1, P_r\}) \rceil$, $R_2 \leq \lceil \frac{1}{2} \log(\min\{P_2, P_r\}) \rceil$,

LD Two-way relay channel

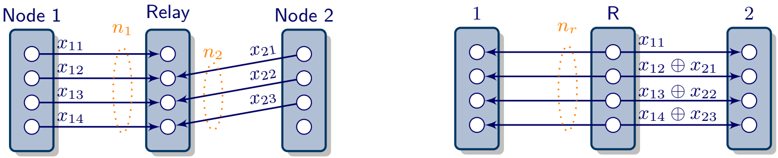


- Achievable rates: $R_1 \leq \min\{n_1, n_r\}$ and $R_2 \leq \min\{n_2, n_r\}$.

$\Rightarrow$ $R_1 \leq \lceil \frac{1}{2} \log(\min\{P_1, P_r\}) \rceil$, $R_2 \leq \lceil \frac{1}{2} \log(\min\{P_2, P_r\}) \rceil$,

- asymmetric rates can be achieved by combining CF and DF

LD Two-way relay channel



- Achievable rates: $R_1 \leq \min\{n_1, n_r\}$ and $R_2 \leq \min\{n_2, n_r\}$.

$\Rightarrow$ $R_1 \leq \lceil \frac{1}{2} \log(\min\{P_1, P_r\}) \rceil$, $R_2 \leq \lceil \frac{1}{2} \log(\min\{P_2, P_r\}) \rceil$,

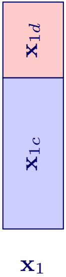
- asymmetric rates can be achieved by combining CF and DF

How to extend to Gaussian two-way relay channels?

CF rate region

Combination of CF and DF

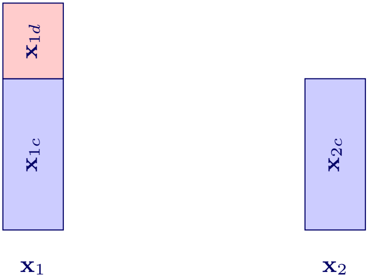
- Node 1: $\mathbf{x}_1 = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}x_{1c}$



CF rate region

Combination of CF and DF

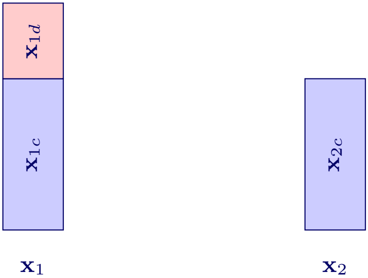
- Node 1: $\mathbf{x}_1 = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}x_{1c}$
- Node 2: $\mathbf{x}_2 = \sqrt{P_c}\mathbf{x}_{2c}$



CF rate region

Combination of CF and DF

- Node 1: $\mathbf{x}_1 = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}x_{1c}$
- Node 2: $\mathbf{x}_2 = \sqrt{P_c}\mathbf{x}_{2c}$
- Powers: $P_c + P_d \leq P_1$, $P_c \leq P_2$.

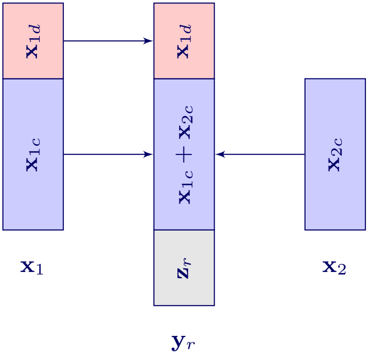


CF rate region

Combination of CF and DF

- Node 1: $\mathbf{x}_1 = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}x_{1c}$
- Node 2: $\mathbf{x}_2 = \sqrt{P_c}\mathbf{x}_{2c}$
- Powers: $P_c + P_d \leq P_1$, $P_c \leq P_2$.
- Relay receives:

$$\mathbf{y}_r = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}(x_{1c} + x_{2c}) + \mathbf{z}_r,$$



CF rate region

Combination of CF and DF

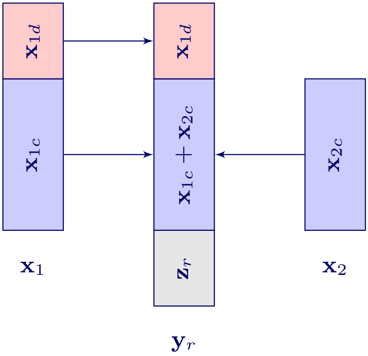
- Node 1: $\mathbf{x}_1 = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}x_{1c}$
- Node 2: $\mathbf{x}_2 = \sqrt{P_c}\mathbf{x}_{2c}$
- Powers: $P_c + P_d \leq P_1,\ P_c \leq P_2.$
- Relay receives:

$$\mathbf{y}_r = \sqrt{P_d}\mathbf{x}_{1d} + \sqrt{P_c}(x_{1c} + x_{2c}) + \mathbf{z}_r,$$

- decodes $\mathbf{x}_{rd} = \mathbf{x}_{1d}$ followed by $\mathbf{x}_{rc} = \mathbf{x}_{1c} + \mathbf{x}_{2c}$,

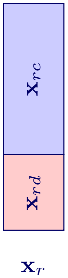
$$R_d \leq \frac{1}{2}\log\left(1 + \frac{P_d}{1 + 2P_c}\right)$$

$$R_c \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + P_c\right)\right]^+$$



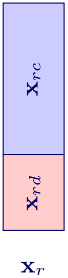
CF rate region

- $\mathbf{x}_r = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd},$



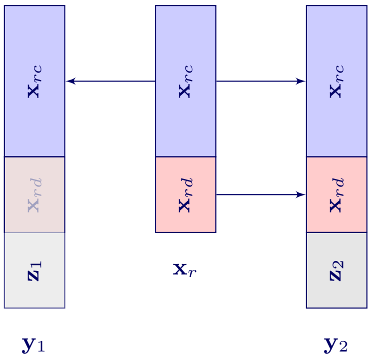
CF rate region

- $\mathbf{x}_r = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd}$,
- Powers: $P_{rc} + P_{rd} \leq P_r$,



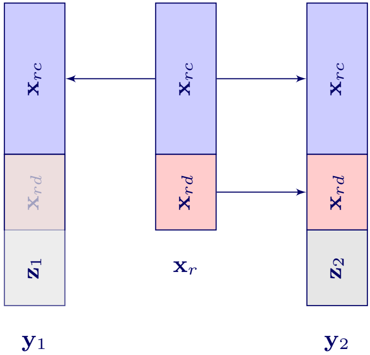
CF rate region

- $\mathbf{x}_r = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd}$,
- Powers: $P_{rc} + P_{rd} \leq P_r$,
- Node i receives
  $\mathbf{y}_i = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd} + \mathbf{z}_i$



CF rate region

- $\mathbf{x}_r = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd}$,
- Powers: $P_{rc} + P_{rd} \leq P_r$,
- Node i receives
  $\mathbf{y}_i = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd} + \mathbf{z}_i$
- both nodes decode $\mathbf{x}_{rc}$, and extract desired CF signal

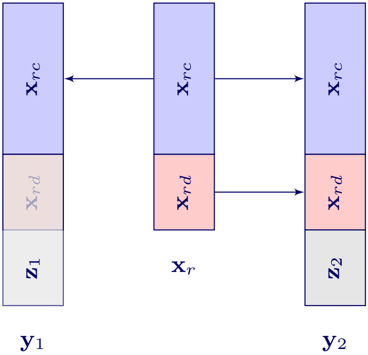


CF rate region

- $\mathbf{x}_r = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd}$,
- Powers: $P_{rc} + P_{rd} \leq P_r$,
- Node i receives
  $\mathbf{y}_i = \sqrt{P_{rc}}\mathbf{x}_{rc} + \sqrt{P_{rd}}\mathbf{x}_{rd} + \mathbf{z}_i$
- both nodes decode $\mathbf{x}_{rc}$, and extract desired CF signal
- and node 2 decodes $\mathbf{x}_{rd}$

$$R_c \leq \frac{1}{2}\log\left(1 + \frac{P_{rc}}{1 + P_{rd}}\right)$$

$$R_d \leq \frac{1}{2}\log\left(1 + P_{rd}\right)$$



CF/DF

Combining CF and DF achieves $R_1 = R_c + R_d$, $R_2 = R_c$, where

$$R_d \leq \min\left\{\frac{1}{2}\log\left(1+\frac{P_d}{1+2P_c}\right), \frac{1}{2}\log\left(1+P_d\right)\right\}$$

$$R_c \leq \min\left\{\left[\frac{1}{2}\log\left(\frac{1}{2}+P_c\right)\right]^+, \frac{1}{2}\log\left(1+\frac{P_c}{1+P_d}\right)\right\}$$

for $P_c + P_d \leq P_1$, $P_c \leq P_2$, $P_{rc} + P_{rd} \leq P_r$.

CF/DF

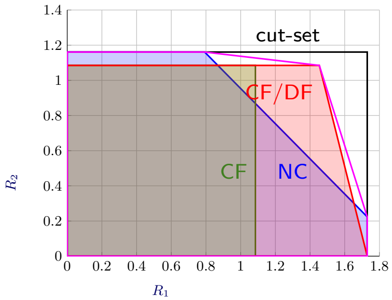
Combining CF and DF achieves $R_1 = R_c + R_d$, $R_2 = R_c$, where

$$R_d \leq \min\left\{\frac{1}{2}\log\left(1+\frac{P_d}{1+2P_c}\right), \frac{1}{2}\log\left(1+P_d\right)\right\}$$

$$R_c \leq \min\left\{\left[\frac{1}{2}\log\left(\frac{1}{2}+P_c\right)\right]^+, \frac{1}{2}\log\left(1+\frac{P_c}{1+P_d}\right)\right\}$$

for $P_c + P_d \leq P_1$, $P_c \leq P_2$, $P_{rc} + P_{rd} \leq P_r$.

Achieves capacity within a constant gap

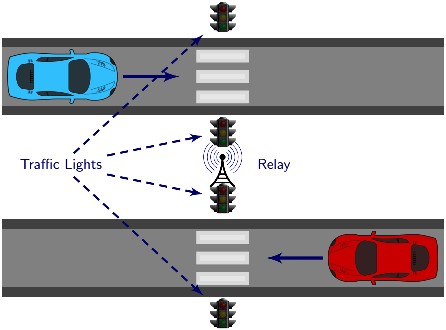


Summary

- Key ingredient: CF using lattice codes (physical-layer network coding)
- Best scheme: Combination of CF, DF, and NC,
- Sum-capacity scales as $\log(P)$, (optimal scaling)

Summary

- Key ingredient: CF using lattice codes (physical-layer network coding)
- Best scheme: Combination of CF, DF, and NC,
- Sum-capacity scales as $\log(P)$, (optimal scaling)
- Consequence: Using a relay as a two-way relay doubles the rate of communication, which is of interest for applications with a delay constraint

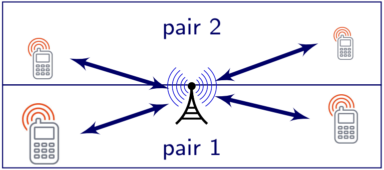


Outline

Multi-pair Two-way Relay Channel

Multiple users communicating pair-wise through a relay [S. *et al.* 09]

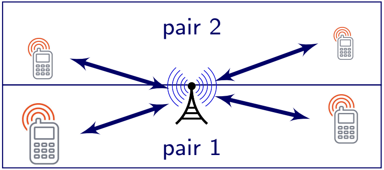
- Combination of CF and DF,



Multi-pair Two-way Relay Channel

Multiple users communicating pair-wise through a relay [S. *et al.* 09]

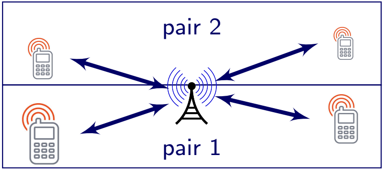
- Combination of CF and DF,
- In each pair (i_k, j_k), one node sends $\mathbf{x}_{i_k d} + \mathbf{x}_{i_k c}$ and the other $\mathbf{x}_{j_k c}$,



Multi-pair Two-way Relay Channel

Multiple users communicating pair-wise through a relay [S. *et al.* 09]

- Combination of CF and DF,
- In each pair (i_k, j_k), one node sends $\mathbf{x}_{i_k d} + \mathbf{x}_{i_k c}$ and the other $\mathbf{x}_{j_k c}$,
- Relay decodes $\mathbf{x}_{i_k d}$ then $\mathbf{x}_{i_k c} + \mathbf{x}_{j_k c}$ of pair k, then pair k'...

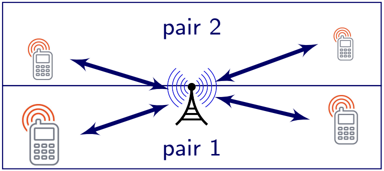


$$R_{kd} \leq \frac{1}{2}\log\left(1 + \frac{P_{kd}}{1 + 2P_{kc} + \sum_{\ell=k+1}^{K}(P_{\ell d} + 2P_{\ell c})}\right)$$

$$R_{kc} \leq \left[\frac{1}{2}\log\left(\frac{1}{2} + \frac{P_{kc}}{1 + \sum_{\ell=k+1}^{K}(P_{\ell d} + 2P_{\ell c})}\right)\right]^{+}$$

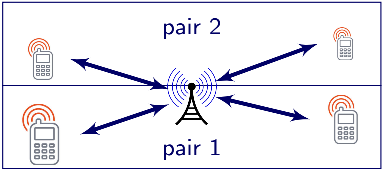
Multi-pair Two-way Relay Channel

- Relay forwards a scaled sum of the decoded signals



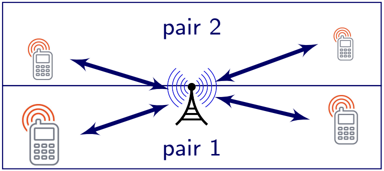
Multi-pair Two-way Relay Channel

- Relay forwards a scaled sum of the decoded signals
- Nodes in pair k decode the signals successively, starting with pair 1 ending with pair k



Multi-pair Two-way Relay Channel

- Relay forwards a scaled sum of the decoded signals
- Nodes in pair k decode the signals successively, starting with pair 1 ending with pair k
- within a constant of the cut-set bound in the Gaussian case,



Remarks

The multi-pair case is similar to the single pair case:

- Sum-rate scaling of $\log(P)$, (optimal scaling)
- Cut-set bounds are nearly tight. Achievability requires:
- CF: Bi-directional communication between two nodes via the relay, and
- DF: Uni-directional communication from one user to the other via the relay.

Remarks

The multi-pair case is similar to the single pair case:

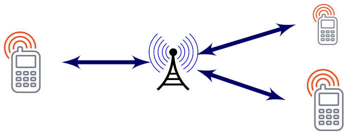
- Sum-rate scaling of $\log(P)$, (optimal scaling)
- Cut-set bounds are nearly tight. Achievability requires:
- CF: Bi-directional communication between two nodes via the relay, and
- DF: Uni-directional communication from one user to the other via the relay.

Do we require new ingredients in multi-user cases?

Outline

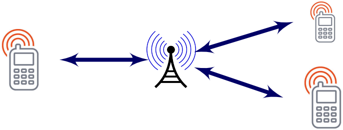
Multi-way Relay Channel

Channel with multiple users communicating in all directions via a relay [Lee & Lim 09]



Multi-way Relay Channel

Channel with multiple users communicating in all directions via a relay [Lee & Lim 09]

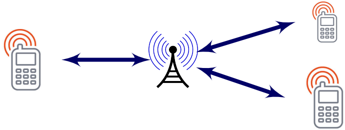


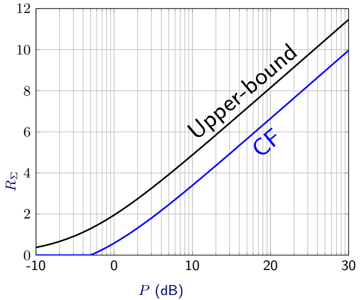
- Cut-set bound scaling of $\frac{3}{2}\log(P)$

Multi-way Relay Channel

Channel with multiple users communicating in all directions via a relay [Lee & Lim 09]

- Cut-set bound scaling of $\frac{3}{2}\log(P)$
- Genie-aided bound scaling of $\log(P)$ [C. & S. 11]

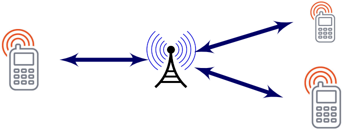


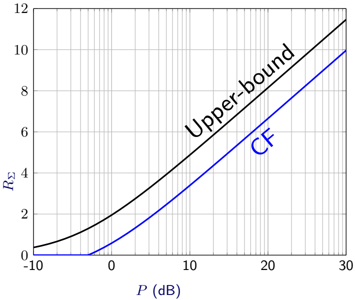


Multi-way Relay Channel

Channel with multiple users communicating in all directions via a relay [Lee & Lim 09]

- Cut-set bound scaling of $\frac{3}{2}\log(P)$
- Genie-aided bound scaling of $\log(P)$ [C. & S. 11]
- Cut-set bounds are not tight!

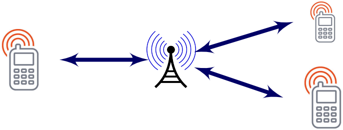


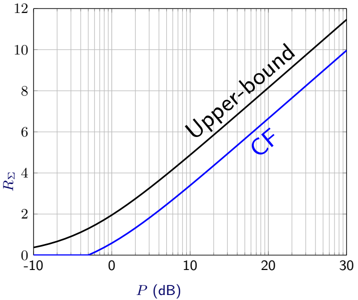


Multi-way Relay Channel

Channel with multiple users communicating in all directions via a relay [Lee & Lim 09]

- Cut-set bound scaling of $\frac{3}{2}\log(P)$
- Genie-aided bound scaling of $\log(P)$ [C. & S. 11]
- Cut-set bounds are not tight!
- CF achieves the optimal scaling





Multi-way Relay Channel

- CF achieves optimal scaling as in the single and multi-pair case

Multi-way Relay Channel

- CF achieves optimal scaling as in the single and multi-pair case
- Capacity region?

Multi-way Relay Channel

- CF achieves optimal scaling as in the single and multi-pair case
- Capacity region?
- Two-way: Bi-directional and uni-directional

Multi-way Relay Channel

- CF achieves optimal scaling as in the single and multi-pair case
- Capacity region?
- Two-way: Bi-directional and uni-directional
- Multi-way: Similar?

Multi-way Relay Channel

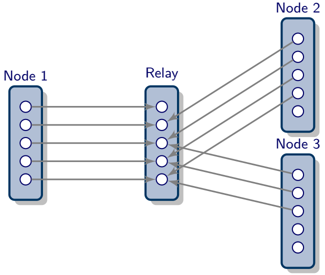
- CF achieves optimal scaling as in the single and multi-pair case
- Capacity region?
- Two-way: Bi-directional and uni-directional
- Multi-way: Similar?
- No!

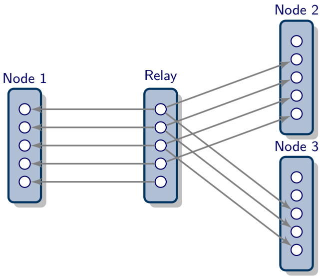
Multi-way Relay Channel

- CF achieves optimal scaling as in the single and multi-pair case
- Capacity region?
- Two-way: Bi-directional and uni-directional
- Multi-way: Similar?
- No!
- Let us check the LD model

Multi-way Relay Channel

Outer bound: Cut-set and genie-aided.

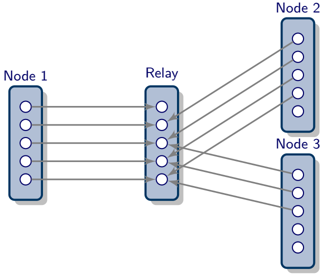


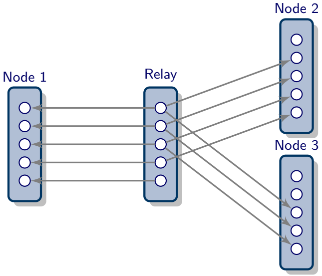


Multi-way Relay Channel

Outer bound: Cut-set and genie-aided.

- Let $n_1 = 5$, $n_2 = 4$, $n_3 = 3$,

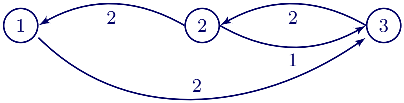


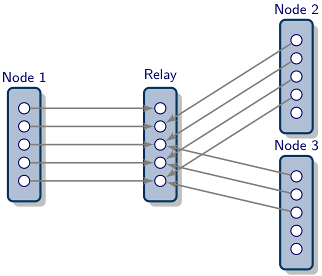


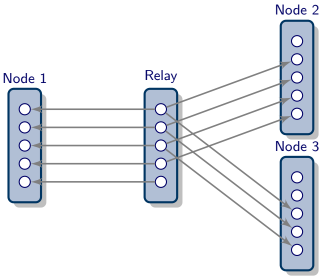
Multi-way Relay Channel

Outer bound: Cut-set and genie-aided.

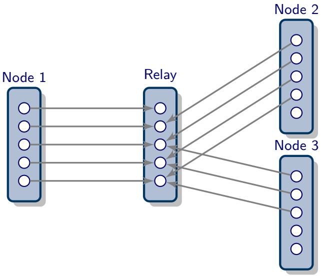
- Let $n_1 = 5$, $n_2 = 4$, $n_3 = 3$,
- $R_{13} = R_{21} = R_{32} = 2$, $R_{23} = 1$





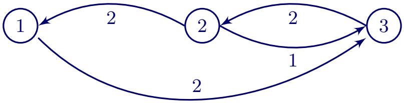


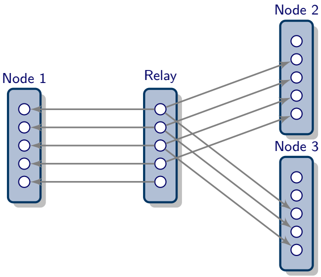
Multi-way Relay Channel



Outer bound: Cut-set and genie-aided.

- Let $n_1 = 5,\ n_2 = 4,\ n_3 = 3,$
- $R_{13} = R_{21} = R_{32} = 2,\ R_{23} = 1$



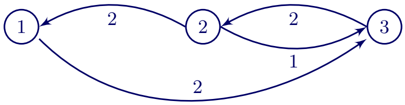


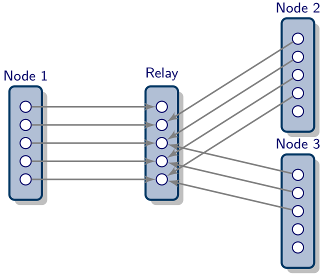
- Rates inside the outer bound

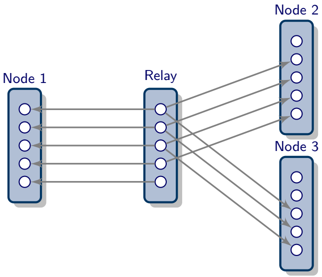
Multi-way Relay Channel

Outer bound: Cut-set and genie-aided.

- Let $n_1 = 5$, $n_2 = 4$, $n_3 = 3$,
- $R_{13} = R_{21} = R_{32} = 2$, $R_{23} = 1$





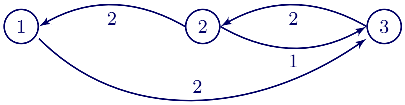


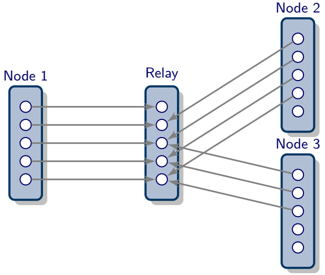
- Rates inside the outer bound
- Achievable?

Multi-way Relay Channel

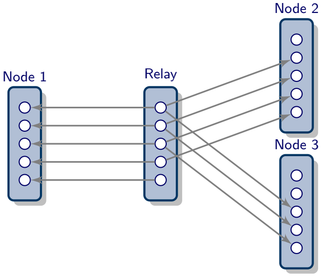
Outer bound: Cut-set and genie-aided.

- Let $n_1 = 5$, $n_2 = 4$, $n_3 = 3$,
- $R_{13} = R_{21} = R_{32} = 2$, $R_{23} = 1$

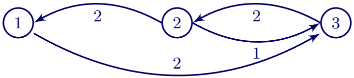




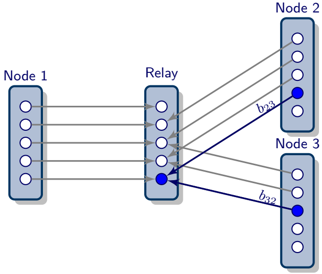
- Rates inside the outer bound
- Achievable?
- Try bi-directional and uni-directional

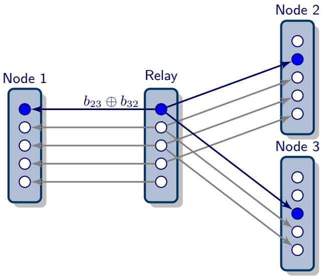


Multi-way Relay Channel

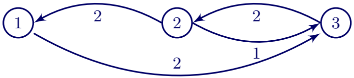


- Bi-directional $2 \leftrightarrow 3$

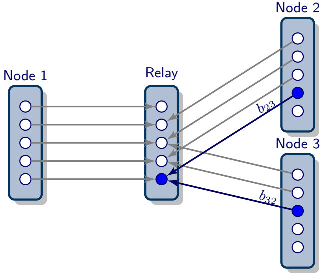


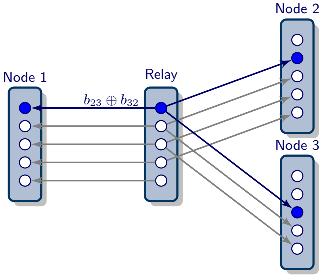


Multi-way Relay Channel

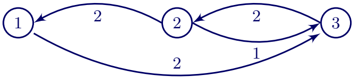


- Bi-directional $2 \leftrightarrow 3$
- Achieves $r_{23} = r_{32} = 1$

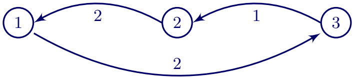


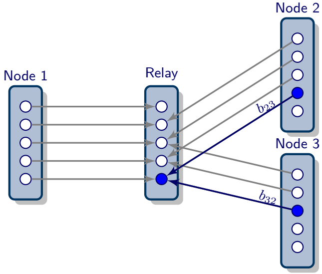


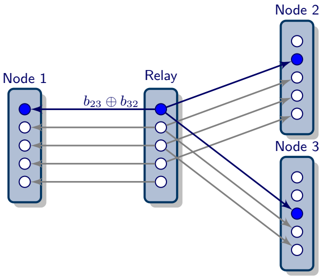
Multi-way Relay Channel



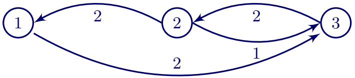
- Bi-directional $2 \leftrightarrow 3$
- Achieves $r_{23} = r_{32} = 1$
- Remainder $R_{13} = R_{21} = 2$, $R_{32} = 1 \Rightarrow$

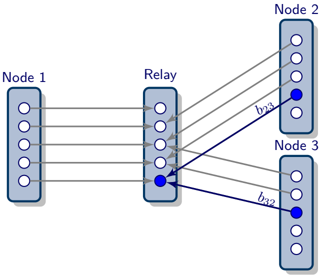




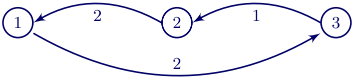


Multi-way Relay Channel

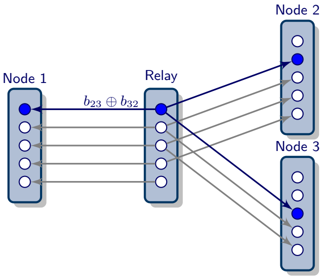




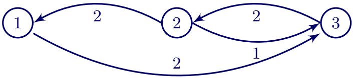
- Bi-directional $2 \leftrightarrow 3$
- Achieves $r_{23} = r_{32} = 1$
- Remainder $R_{13} = R_{21} = 2$, $R_{32} = 1 \Rightarrow$



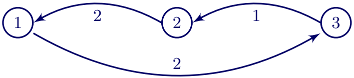
- Uni-directional $1 \rightarrow 3$, $2 \rightarrow 1$, and $3 \rightarrow 2$ requires 5 bit-pipes



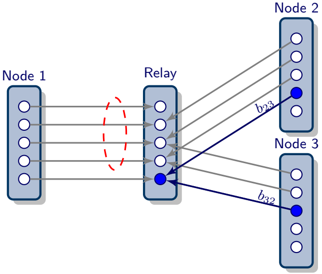
Multi-way Relay Channel

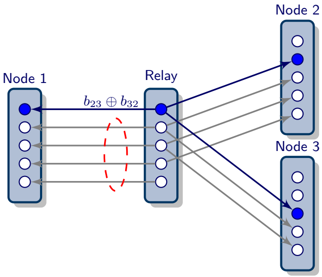


- Bi-directional $2 \leftrightarrow 3$
- Achieves $r_{23} = r_{32} = 1$
- Remainder $R_{13} = R_{21} = 2$, $R_{32} = 1 \Rightarrow$

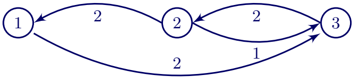


- Uni-directional $1 \rightarrow 3$, $2 \rightarrow 1$, and $3 \rightarrow 2$ requires 5 bit-pipes
- Relay has only 4 remaining!

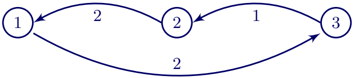




Multi-way Relay Channel

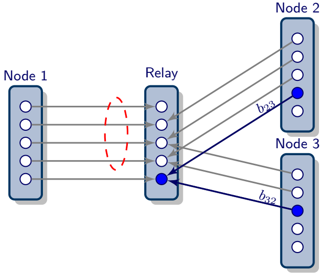


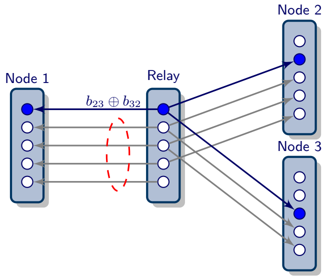
- Bi-directional $2 \leftrightarrow 3$
- Achieves $r_{23} = r_{32} = 1$
- Remainder $R_{13} = R_{21} = 2$, $R_{32} = 1 \Rightarrow$



- Uni-directional $1 \to 3$, $2 \to 1$, and $3 \to 2$ requires 5 bit-pipes
- Relay has only 4 remaining!

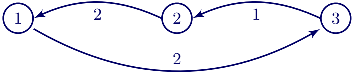
$\Rightarrow$ Achievability requires more CF

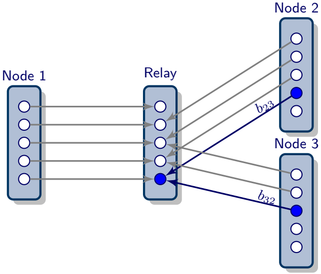


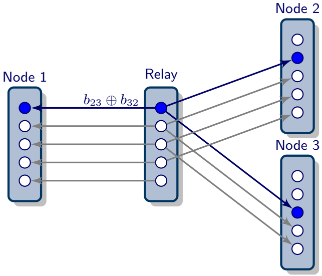


Multi-way Relay Channel

- Bi-directional $2 \leftrightarrow 3$

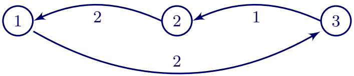




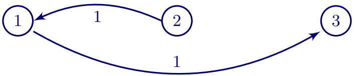


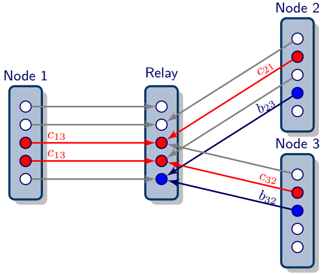
Multi-way Relay Channel

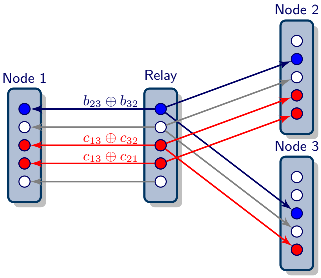
- Bi-directional $2 \leftrightarrow 3$



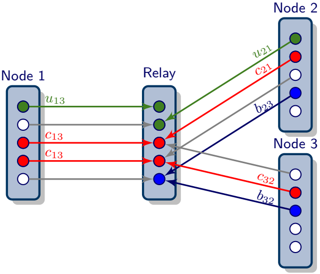
- Cyclic $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$:
- Achieves $r_{13} = r_{32} = r_{21} = 1$
- Remainder $R_{13} = R_{21} = 1$



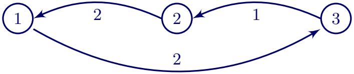




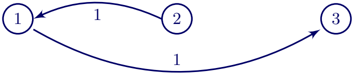
Multi-way Relay Channel

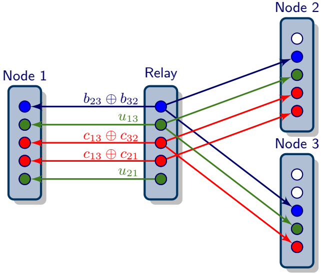


- Bi-directional $2 \leftrightarrow 3$



- Cyclic $1 \to 3 \to 2 \to 1$:
- Achieves $r_{13} = r_{32} = r_{21} = 1$
- Remainder $R_{13} = R_{21} = 1$

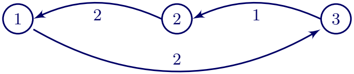


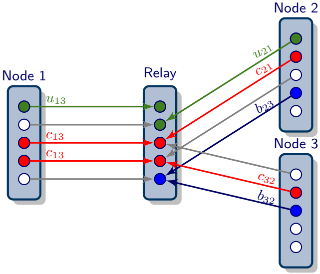


- Uni-directional $1 \to 3$ and $2 \to 1$: requires 2 bit-pipes
- Relay has 2 remaining

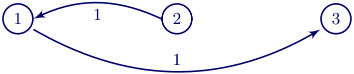
Multi-way Relay Channel

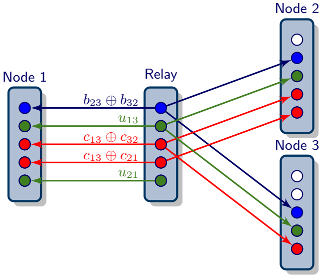
- Bi-directional $2 \leftrightarrow 3$





- Cyclic $1 \to 3 \to 2 \to 1$:
- Achieves $r_{13} = r_{32} = r_{21} = 1$
- Remainder $R_{13} = R_{21} = 1$

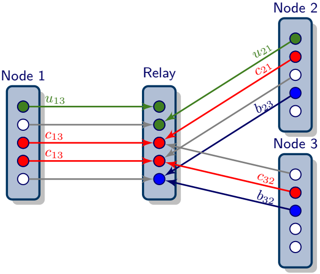


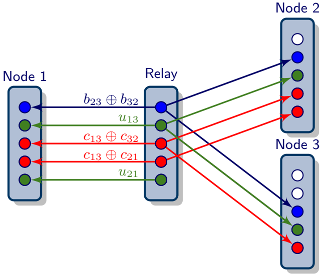


- Uni-directional $1 \to 3$ and $2 \to 1$: requires 2 bit-pipes
- Relay has 2 remaining
- Desired rate achieved!

Multi-way Relay Channel

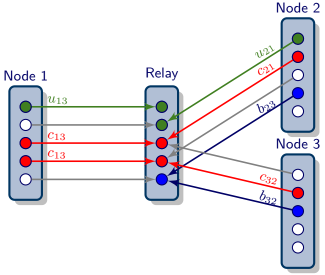
- Additional ingredient: Cyclic Communication

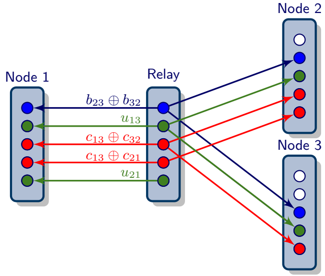




Multi-way Relay Channel

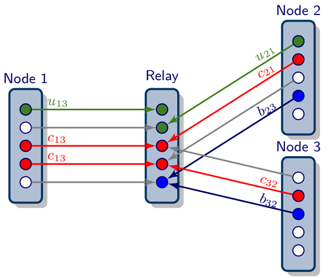
- Additional ingredient: Cyclic Communication
- Remark:
  Bi-directional: 2 bits per bit-pipe
  Cyclic: 3/2 bits per bit-pipe
  Uni-directional: 1 bit per bit-pipe

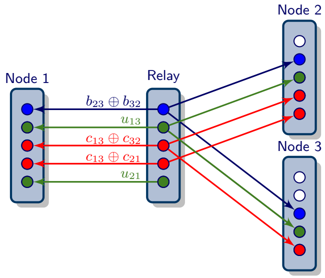




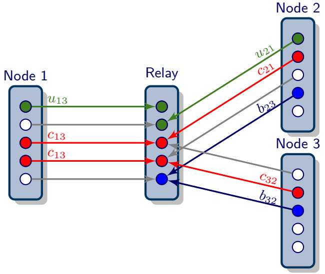
Multi-way Relay Channel

- Additional ingredient: Cyclic Communication
- Remark:
  Bi-directional: 2 bits per bit-pipe
  Cyclic: 3/2 bits per bit-pipe
  Uni-directional: 1 bit per bit-pipe
- Best scheme: Combination of the three

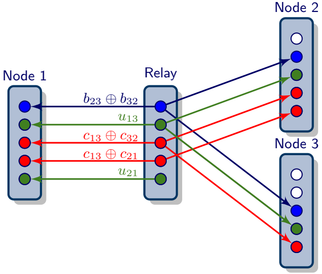




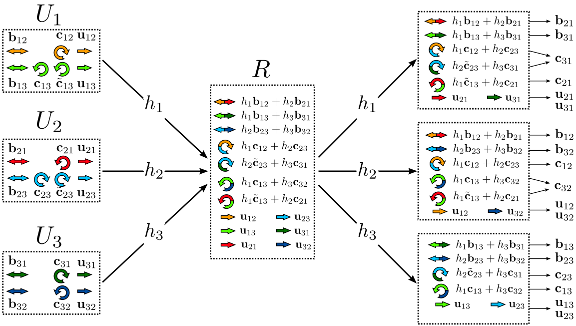
Multi-way Relay Channel



- Additional ingredient: Cyclic Communication
- Remark:
  Bi-directional: 2 bits per bit-pipe
  Cyclic: 3/2 bits per bit-pipe
  Uni-directional: 1 bit per bit-pipe
- Best scheme: Combination of the three
- LD case: Capacity achieving [C. & S. 11]



Multi-way Relay Channel

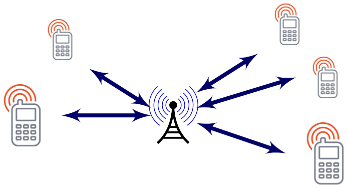


Gaussian case:

- node i sends $\mathbf{x}_{ib} + \mathbf{x}_{ic} + \mathbf{x}_{iu}$ (bi-directional, cyclic, uni-directional)
- relay computes the sum of bi-directional signals, cyclic signals, and decodes the uni-directional ones
- nodes decode successively and obtain their desired signals
- Problem reduces to power allocation (near optimal allocation in [C. & S. 12])

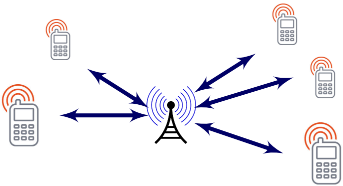
K-user case

- Sum-capacity upper bound scales as $\log(P)$,



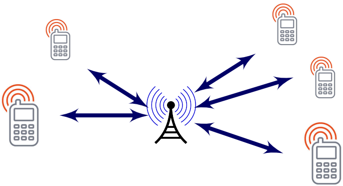
K-user case

- Sum-capacity upper bound scales as $\log(P)$,
- Simple scheduling: Schedule one pair of users at a time
- Channel reduces to a sequence of two-way relay channels



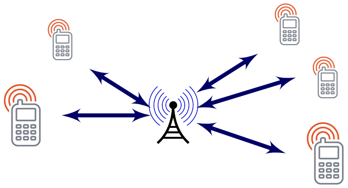
K-user case

- Sum-capacity upper bound scales as $\log(P)$,
- Simple scheduling: Schedule one pair of users at a time
- Channel reduces to a sequence of two-way relay channels
- Apply bi-directional communication over each two-way relay channel



K-user case

- Sum-capacity upper bound scales as $\log(P)$,
- Simple scheduling: Schedule one pair of users at a time
- Channel reduces to a sequence of two-way relay channels
- Apply bi-directional communication over each two-way relay channel
- achieves sum-capacity within a constant gap

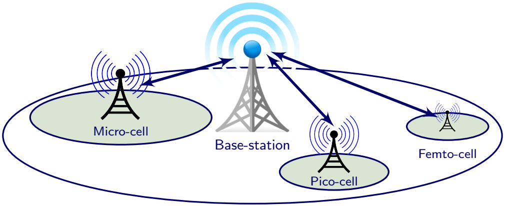


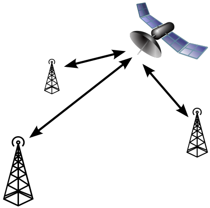
Summary

- Key ingredient: CF for bi-directional and cyclic communication
- Best scheme: Combination of bi-directional, cyclic, and uni-directional
- Sum-capacity scales as $\log(P)$,

Summary

- Key ingredient: CF for bi-directional and cyclic communication
- Best scheme: Combination of bi-directional, cyclic, and uni-directional
- Sum-capacity scales as $\log(P)$,
- Consequence: Treating different modes of information flow differently increases the communication rate



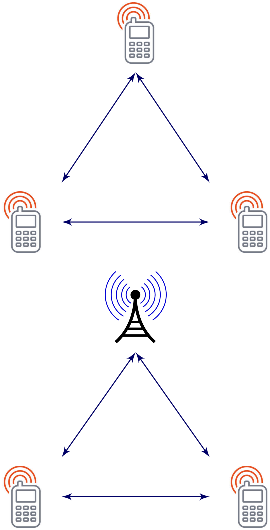


Outline

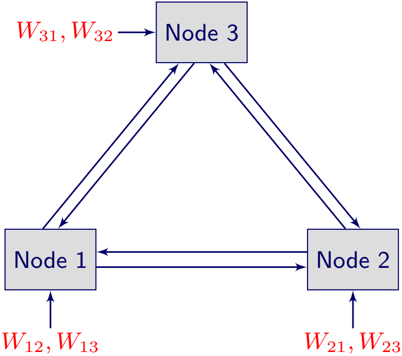
1. Two-way channel
2. Two-way relay channel
   - The linear-deterministic approximation
   - Lattice codes
3. Multi-way relay channel
   - Multi-pair Two-way Relay Channel
   - Multi-way Relay Channel
4. **Multi-way Channel**

3-Way Channel

- 3 (or more) nodes communicating with each other in multiple directions
- Extension of Shannon's two-way channel
- A suitable model for D2D systems (offloading traffic from the cellular network [Asadi *et al.*])

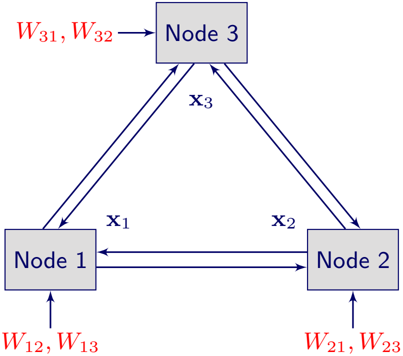


3-Way Channel (3WC)



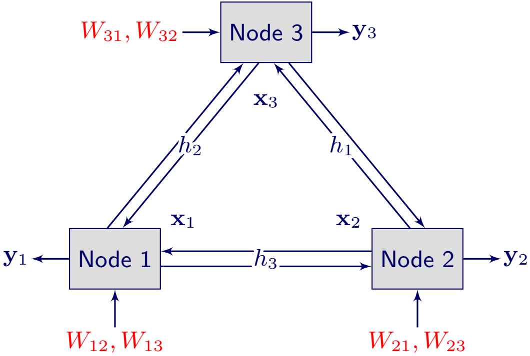
- Full message-exchange: Message W_{ij} from node i to j,

3-Way Channel (3WC)



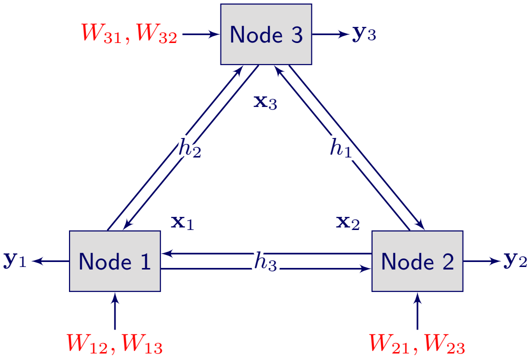
- Full message-exchange: Message W_{ij} from node i to j,
- Tx signal: $\mathbf{x}_j$, power P,

3-Way Channel (3WC)



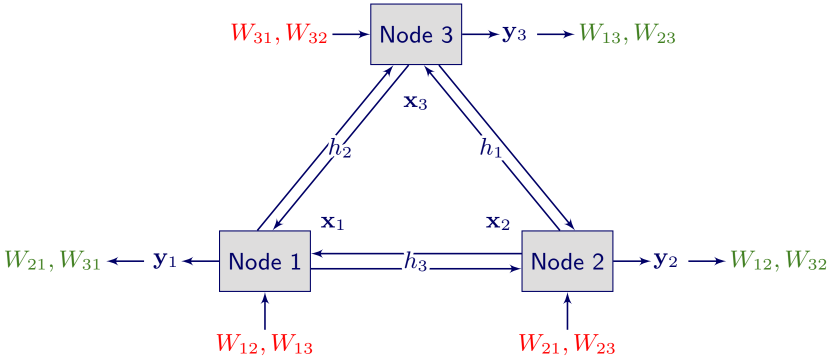
- Full message-exchange: Message W_{ij} from node i to j,
- Tx signal: $\mathbf{x}_j$, power P,
- Rx signal: $\mathbf{y}_k = h_i\mathbf{x}_j + h_j\mathbf{x}_i + \mathbf{z}_k$ (reciprocal channels, unit noise power)

3-Way Channel (3WC)



- Full message-exchange: Message W_{ij} from node i to j,
- Tx signal: $\mathbf{x}_j$, power P,
- Rx signal: $\mathbf{y}_k = h_i\mathbf{x}_j + h_j\mathbf{x}_i + \mathbf{z}_k$ (reciprocal channels, unit noise power)
- w.l.o.g. $h_3^2 \geq h_2^2 \geq h_1^2$,

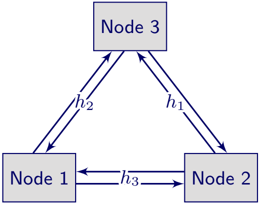
3-Way Channel (3WC)



- Full message-exchange: Message W_{ij} from node i to j,
- Tx signal: $\mathbf{x}_j$, power P,
- Rx signal: $\mathbf{y}_k = h_i \mathbf{x}_j + h_j \mathbf{x}_i + \mathbf{z}_k$ (reciprocal channels, unit noise power)
- w.l.o.g. $h_3^2 \geq h_2^2 \geq h_1^2$,
- Node k decodes W_{ik} and W_{jk},

Sum-Capacity

- Two-way channel: Cut-set bound tight, capacity scales as $\log(P)$,
- 3-way channel: Cut-set bound not tight, capacity also scales as $\log(P)$



Sum-capacity

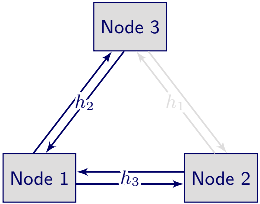
The sum-capacity of the 3-way channel is bounded by

$$\log(1 + h_3^2 P) \leq C_\Sigma \leq \log(1 + h_3^2 P) + 2.$$

- Converse: Genie-aided bound [C. *et al.* 14]
- Achievability: Only users 1 and 2 communicate
- Optimal scaling can also be achieved by scheduling

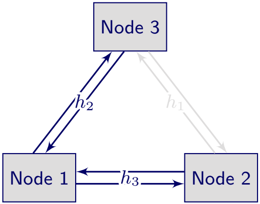
Capacity Region

- Two-way channel scheme suffices for sum-capacity,



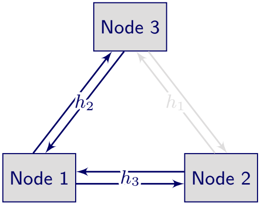
Capacity Region

- Two-way channel scheme suffices for sum-capacity,
- but not for Capacity region,



Capacity Region

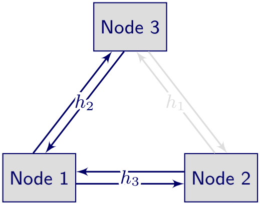
- Two-way channel scheme suffices for sum-capacity,
- but not for Capacity region,



- Assume nodes 2 & 3 want to communicate, but $h_1^2 \ll 1$

Capacity Region

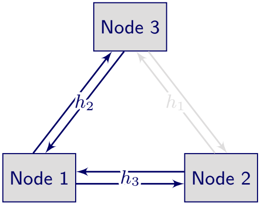
- Two-way channel scheme suffices for sum-capacity,
- but not for Capacity region,



- Assume nodes 2 & 3 want to communicate, but $h_1^2 \ll 1$
- Communication still possible via node 1 as a relay (two-way relay channel)

Capacity Region

- Two-way channel scheme suffices for sum-capacity,
- but not for Capacity region,

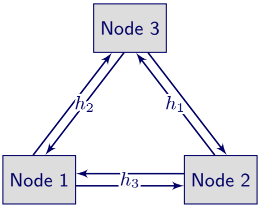


- Assume nodes 2 & 3 want to communicate, but $h_1^2 \ll 1$
- Communication still possible via node 1 as a relay (two-way relay channel)
- Relaying is necessary for capacity region!

Capacity Region

How to find the capacity region?

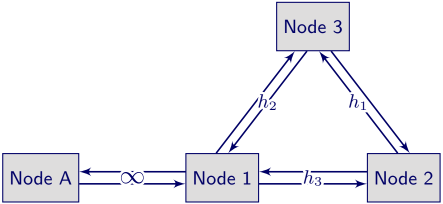
- Trick: Transform the channel into a Y-channel!



Capacity Region

How to find the capacity region?

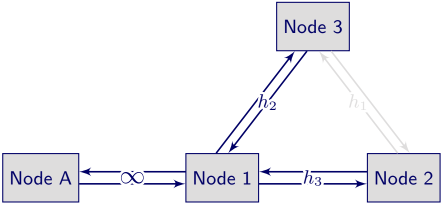
- Trick: Transform the channel into a Y-channel!
- Split stronger node into two,



Capacity Region

How to find the capacity region?

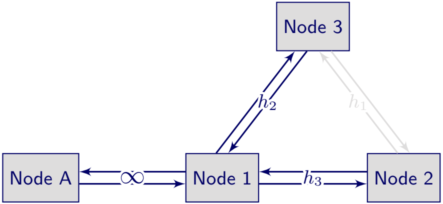
- Trick: Transform the channel into a Y-channel!
- Split stronger node into two,
  - Assume $h_1^2 = 0 \Rightarrow$ Y-channel!



Capacity Region

How to find the capacity region?

- Trick: Transform the channel into a Y-channel!
- Split stronger node into two,



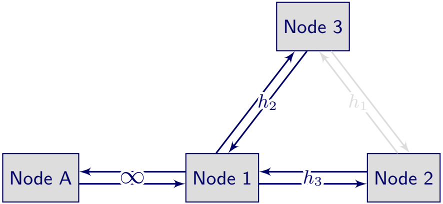
  - Assume $h_1^2 = 0 \Rightarrow$ Y-channel!

$\Rightarrow$ Capacity achieving scheme for the Y-channel is capacity achieving for the 3-way channel

Capacity Region

How to find the capacity region?

- Trick: Transform the channel into a Y-channel!
- Split stronger node into two,



- Assume $h_1^2 = 0 \Rightarrow$ Y-channel!

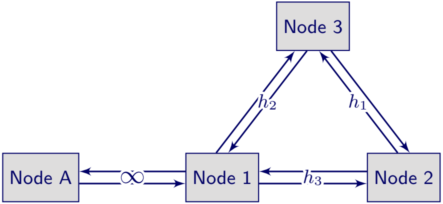
$\Rightarrow$ Capacity achieving scheme for the Y-channel is capacity achieving for the 3-way channel

- What if $h_1^2 > 0$?

Capacity Region

If $h_1^2 > 0$:

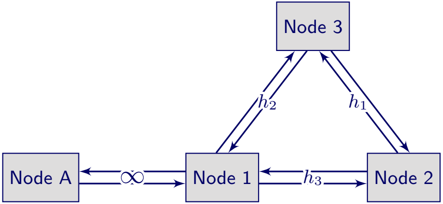
- Interference between nodes 2 and 3 (w.r.t. Y-channel scheme)



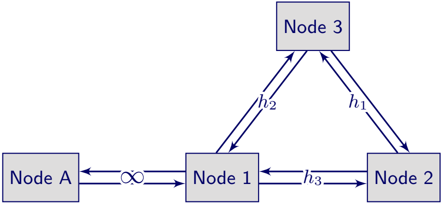
Capacity Region

If $h_1^2 > 0$:

- Interference between nodes 2 and 3 (w.r.t. Y-channel scheme)
- How to resolve interference?



Capacity Region



If $h_1^2 > 0$:

- Interference between nodes 2 and 3 (w.r.t. Y-channel scheme)
- How to resolve interference?

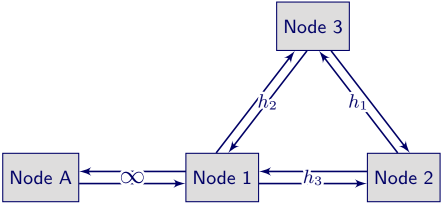
If interference at 3 is:

- A desired signal at 3: Backward decoding:

$$\mathbf{y}_3(B) = h_2\mathbf{x}_1(B) + h_1\mathbf{x}_{23}(B) + \mathbf{z}_3(B), \quad \mathbf{y}_3(B+1) = h_2\mathbf{x}_1(B+1) + \mathbf{z}_3(B+1)$$

- After decoding desired signals from $\mathbf{x}_1(B+1)$, node 3 removes interference from $\mathbf{x}_{23}(B)$ (Causality)

Capacity Region



If $h_1^2 > 0$:

- Interference between nodes 2 and 3 (w.r.t. Y-channel scheme)
- How to resolve interference?

If interference at 3 is:

- A desired signal at 3: Backward decoding:

$$\mathbf{y}_3(B) = h_2\mathbf{x}_1(B) + h_1\mathbf{x}_{23}(B) + \mathbf{z}_3(B), \quad \mathbf{y}_3(B+1) = h_2\mathbf{x}_1(B+1) + \mathbf{z}_3(B+1)$$

- After decoding desired signals from $\mathbf{x}_1(B+1)$, node 3 removes interference from $\mathbf{x}_{23}(B)$ (Causality)
- A desired signal at 1: Interference neutralization

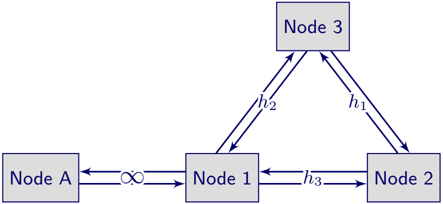
$$\mathbf{y}_3 = h_2\mathbf{x}_1 + h_1\mathbf{x}_{21} + \mathbf{z}_3$$

- Node 2 pre-transmits a signal for interference neutralization: $\mathbf{x}_1 = \mathbf{x}_1' - \frac{h_1}{h_2}\mathbf{x}_{21}$

Capacity Region

Main ingredients

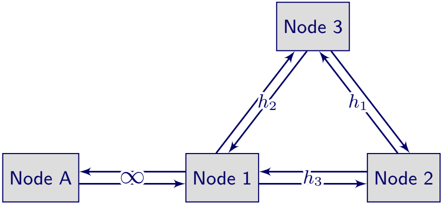
- Y-channel scheme: Bi-directional, cyclic, and uni-directional communication



Capacity Region

Main ingredients

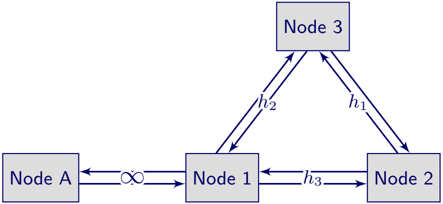
- Y-channel scheme: Bi-directional, cyclic, and uni-directional communication
- For resolving interference between nodes 2 and 3: Backward decoding and interference neutralization



Capacity Region

Main ingredients

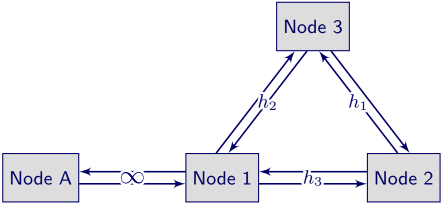
- Y-channel scheme: Bi-directional, cyclic, and uni-directional communication
- For resolving interference between nodes 2 and 3: Backward decoding and interference neutralization
    - Outer bound: Genie-aided and cut-set



Capacity Region

Main ingredients

- Y-channel scheme: Bi-directional, cyclic, and uni-directional communication
- For resolving interference between nodes 2 and 3: Backward decoding and interference neutralization

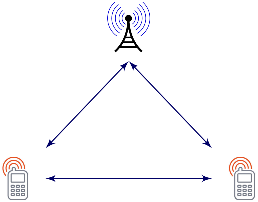


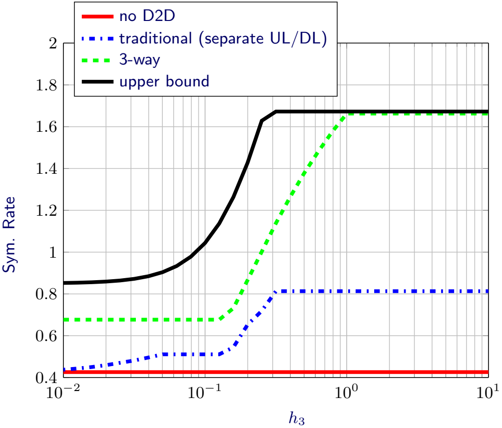
- Outer bound: Genie-aided and cut-set
- Capacity region of the LD case, and approximate capacity of the Gaussian case [C *et al.* 14],

Application

D2D communications:

Impact of (D2D) channel





Part 3: MIMO

Outline

From Capacity to Capacity Region

Single-user (MIMO P2P):

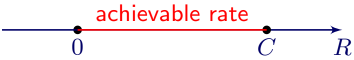
Input covariance $\mathbf{Q}$, tr$(\mathbf{Q}) \leq P$

From Capacity to Capacity Region

Single-user (MIMO P2P):

Input covariance $\mathbf{Q}$, tr$(\mathbf{Q}) \leq P$

Capacity: $C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$



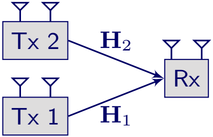
From Capacity to Capacity Region

Single-user (MIMO P2P):

Capacity: $C = \log|\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$

Input covariance $\mathbf{Q}$, $\text{tr}(\mathbf{Q}) \leq P$

Multi-user (MIMO MAC):



Input covariance $\mathbf{Q}_i$, $i = 1, 2$,
$\text{tr}(\mathbf{Q}_i) \leq P$

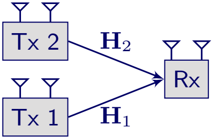
From Capacity to Capacity Region

Single-user (MIMO P2P):

Input covariance $\mathbf{Q}$, $\text{tr}(\mathbf{Q}) \leq P$

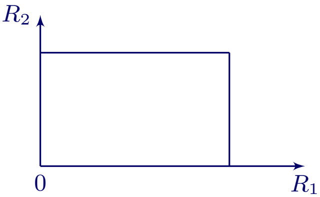
Capacity: $C = \log |\mathbf{I} + \mathbf{HQH}^H|$

Multi-user (MIMO MAC):



Input covariance $\mathbf{Q}_i$, $i = 1, 2$,
$\text{tr}(\mathbf{Q}_i) \leq P$

Capacity region:
$R_i \leq \log |\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$,



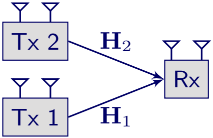
From Capacity to Capacity Region

Single-user (MIMO P2P):

Input covariance $\mathbf{Q}$, tr$(\mathbf{Q}) \leq P$

Capacity: $C = \log|\mathbf{I} + \mathbf{HQH}^H|$

Multi-user (MIMO MAC):

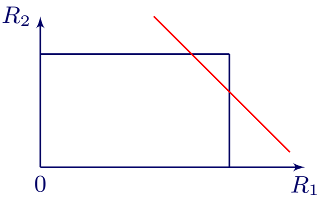


Input covariance $\mathbf{Q}_i$, $i = 1, 2$,
tr$(\mathbf{Q}_i) \leq P$

Capacity region:

$R_i \leq \log|\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$,

$R_1 + R_2 \leq \log|\mathbf{I} + \mathbf{H}_1\mathbf{Q}_1\mathbf{H}_1^H + \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^H|$



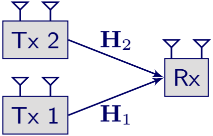
From Capacity to Capacity Region

Single-user (MIMO P2P):

Input covariance $\mathbf{Q}$, tr$(\mathbf{Q}) \leq P$

Capacity: $C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$

Multi-user (MIMO MAC):

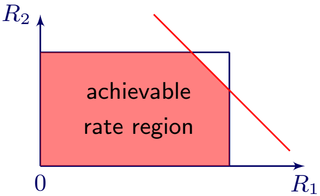


Input covariance $\mathbf{Q}_i$, $i = 1, 2$, tr$(\mathbf{Q}_i) \leq P$

Capacity region:

$R_i \leq \log |\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$,

$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1\mathbf{Q}_1\mathbf{H}_1^H + \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^H|$



Capacity to DoF

Single-user (MIMO P2P):

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$$

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$$

Optimization: water-filling

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$$

Optimization: water-filling

DoF:

- M Tx antennas, N Rx antennas

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^H|$$

Optimization: water-filling

DoF:

- M Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H} \in \mathbb{C}^{N\times M} \Rightarrow$ rank$(\mathbf{H}) = \min\{M, N\}$

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{HQH}^H|$$

Optimization: water-filling

DoF:

- M Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H} \in \mathbb{C}^{N \times M} \Rightarrow \text{rank}(\mathbf{H}) = \min\{M, N\}$

$\Rightarrow$ $C \approx \min\{M, N\} C(P)$ at high P

$C(P)$: SISO P2P capacity

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log |\mathbf{I} + \mathbf{HQH}^H|$$

Optimization: water-filling

DoF:

- M Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H} \in \mathbb{C}^{N\times M} \Rightarrow$ rank$(\mathbf{H}) = \min\{M, N\}$

$\Rightarrow$ $C \approx \min\{M, N\}C(P)$ at high P $\qquad C(P)$: SISO P2P capacity

- DoF: $d = \lim_{P\to\infty} \frac{\text{Capacity}}{C(P)} =$ rank$(\mathbf{H})$ $\qquad \Rightarrow d = \min\{M, N\}$

Capacity to DoF

Single-user (MIMO P2P):

Capacity:

$$C = \log\left|\mathbf{I} + \mathbf{HQH}^H\right|$$

Optimization: water-filling

DoF:

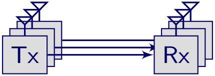
- M Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H} \in \mathbb{C}^{N\times M} \Rightarrow$ rank$(\mathbf{H}) = \min\{M, N\}$

$\Rightarrow$ $C \approx \min\{M, N\}C(P)$ at high P $C(P)$: SISO P2P capacity

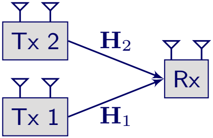
- DoF: $d = \lim_{P\to\infty} \frac{\text{Capacity}}{C(P)} =$ rank$(\mathbf{H})$ $\Rightarrow d = \min\{M, N\}$

$\Rightarrow$ Capacity equivalent to that of d parallel SISO P2P channels!



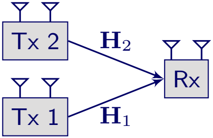
Capacity to DoF

Multi-user (MIMO MAC):



Capacity to DoF

Multi-user (MIMO MAC):



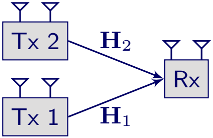
Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

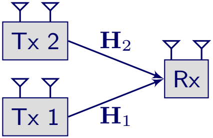
$$R_i \leq \log|\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log|\mathbf{I} + \mathbf{H}_1\mathbf{Q}_1\mathbf{H}_1^H + \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^H|$$

Optimization: Iterative water-filling

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

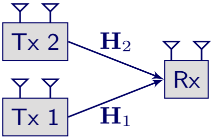
Optimization: Iterative water-filling

DoF:

- M_i Tx antennas, N Rx antennas

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

Optimization: Iterative water-filling

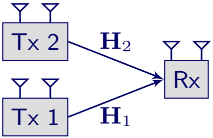
DoF:

- M_i Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H}_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(\mathbf{H}_i) = \min\{M_i, N\}$

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

Optimization: Iterative water-filling

DoF:

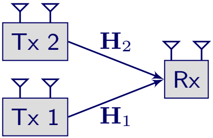
- M_i Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H}_i \in \mathbb{C}^{N \times M_i} \Rightarrow$ rank$(\mathbf{H}_i) = \min\{M_i, N\}$

- $C_\Sigma \approx \min\{M_1 + M_2, N\} C(P)$ at high P

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

Optimization: Iterative water-filling

DoF:

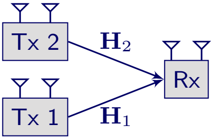
- M_i Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H}_i \in \mathbb{C}^{N \times M_i} \Rightarrow$ rank$(\mathbf{H}_i) = \min\{M_i, N\}$

- $C_\Sigma \approx \min\{M_1 + M_2, N\} C(P)$ at high P
- DoF: $d_i = \lim_{P \to \infty} \frac{R_i}{C(P)}$

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log|\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log|\mathbf{I} + \mathbf{H}_1\mathbf{Q}_1\mathbf{H}_1^H + \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^H|$$

Optimization: Iterative water-filling

DoF:

- M_i Tx antennas, N Rx antennas

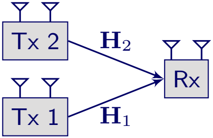
$\Rightarrow$ $\mathbf{H}_i \in \mathbb{C}^{N\times M_i} \Rightarrow$ rank$(\mathbf{H}_i) = \min\{M_i, N\}$

- $C_\Sigma \approx \min\{M_1 + M_2, N\}C(P)$ at high P
- DoF: $d_i = \lim_{P\to\infty} \frac{R_i}{C(P)}$

$\Rightarrow$ DoF region: $d_i \leq$ rank$(\mathbf{H}_i)$, $d_1 + d_2 \leq$ rank$([\mathbf{H}_1,\ \mathbf{H}_2])$

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log|\mathbf{I} + \mathbf{H}_i\mathbf{Q}_i\mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log|\mathbf{I} + \mathbf{H}_1\mathbf{Q}_1\mathbf{H}_1^H + \mathbf{H}_2\mathbf{Q}_2\mathbf{H}_2^H|$$

Optimization: Iterative water-filling

DoF:

- M_i Tx antennas, N Rx antennas

$\Rightarrow$ $\mathbf{H}_i \in \mathbb{C}^{N\times M_i} \Rightarrow$ rank$(\mathbf{H}_i) = \min\{M_i, N\}$

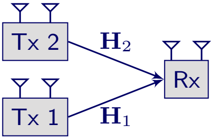
- $C_\Sigma \approx \min\{M_1 + M_2, N\}C(P)$ at high P
- DoF: $d_i = \lim_{P\to\infty} \frac{R_i}{C(P)}$

$\Rightarrow$ DoF region: $d_i \leq$ rank$(\mathbf{H}_i)$, $d_1 + d_2 \leq$ rank$([\mathbf{H}_1,\ \mathbf{H}_2])$

$\Rightarrow$ $d_\Sigma = \min\{M_1 + M_2, N\}$

Capacity to DoF

Multi-user (MIMO MAC):



Capacity region:

$$R_i \leq \log |\mathbf{I} + \mathbf{H}_i \mathbf{Q}_i \mathbf{H}_i^H|$$

$$R_1 + R_2 \leq \log |\mathbf{I} + \mathbf{H}_1 \mathbf{Q}_1 \mathbf{H}_1^H + \mathbf{H}_2 \mathbf{Q}_2 \mathbf{H}_2^H|$$

Optimization: Iterative water-filling

DoF:

- M_i Tx antennas, N Rx antennas

⇒ $\mathbf{H}_i \in \mathbb{C}^{N \times M_i} \Rightarrow$ rank$(\mathbf{H}_i) = \min\{M_i, N\}$

- $C_\Sigma \approx \min\{M_1 + M_2, N\} C(P)$ at high P
- DoF: $d_i = \lim_{P \to \infty} \frac{R_i}{C(P)}$

⇒ DoF region: $d_i \leq$ rank$(\mathbf{H}_i)$, $d_1 + d_2 \leq$ rank$([\mathbf{H}_1,\ \mathbf{H}_2])$

⇒ $d_\Sigma = \min\{M_1 + M_2, N\}$

⇒ Sum-capacity equivalent to that of d_Σ parallel SISO P2P channels!

DoF

DoF

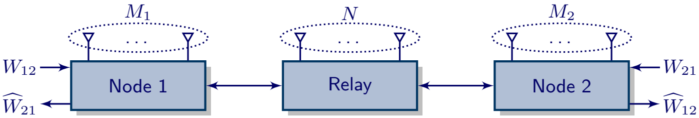
DoF d can be interpreted as the number of parallel streams that can be sent simultaneously over a channel.
It leads to a capacity approximation as

$$C_\Sigma = dC(P) + o\left(C(P)\right),$$

($C(P)$: capacity of a P2P channel)

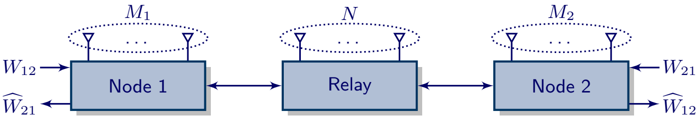
Outline

MIMO Two-Way Relay Channel



DoF characterization [Gündüz *et al.* 08]

MIMO Two-Way Relay Channel



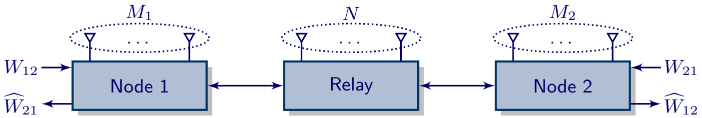
Cut-set bound:

- Node i can not send more than M_i streams,
- Node i can not receive more than M_i streams,
- Relay can relay at most $2N$ streams (PLNC gain),

$\Rightarrow$ Total streams $2\min\{M_1, M_2, N\}$ DoF

- achievable by Compress-forward e.g. [Gündüz *et al.* 08]

MIMO Two-Way Relay Channel



Cut-set bound:

- Node i can not send more than M_i streams,
- Node i can not receive more than M_i streams,
- Relay can relay at most $2N$ streams (PLNC gain),

$\Rightarrow$ Total streams $2\min\{M_1, M_2, N\}$ DoF

- achievable by Compress-forward e.g. [Gündüz *et al.* 08]
- Next: Simple achievability scheme

Simple Achievability

Main Ingredients:

- Channel diagonalization
- Signal alignment

Outline

1. From Capacity to DoF
2. MIMO Two-Way Relay Channel
   - Channel diagonalization
   - Signal Alignment
3. MIMO multi-way relay channel
   - Sum-DoF
   - DoF Region
4. MIMO Multi-way Channel

Channel Diagonalization

Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

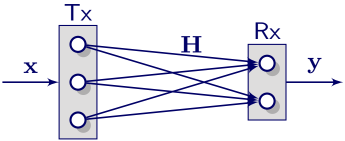
Channel Diagonalization

Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:



Channel Diagonalization

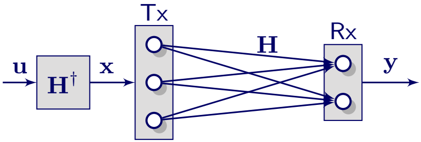
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

- Pseudo-inverse:
  $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$



Channel Diagonalization

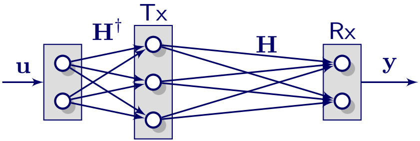
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



Channel Diagonalization

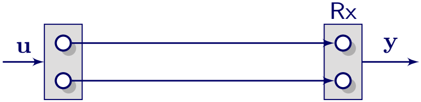
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



N parallel SISO P2P channels!

Channel Diagonalization

Definition (Channel diagonlization)

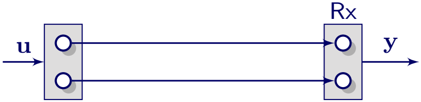
Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

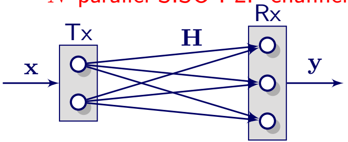
$M \geq N$: ZF pre-coding:

- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$

$M \leq N$: ZF post-coding:



N parallel SISO P2P channels!



Channel Diagonalization

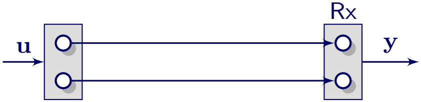
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

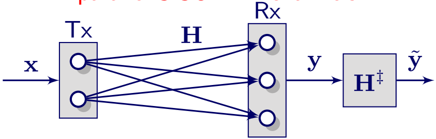
- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



N parallel SISO P2P channels!

$M \leq N$: ZF post-coding:

- Pseudo-inverse: $\mathbf{H}^{\ddagger} = [\mathbf{H}^{H}\mathbf{H}]^{-1}\mathbf{H}^{H}$



Channel Diagonalization

Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

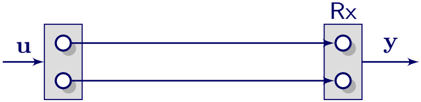
MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

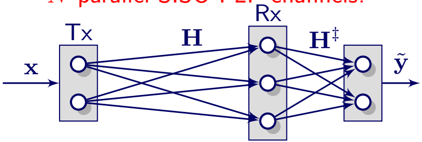
- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$

$M \leq N$: ZF post-coding:

- Pseudo-inverse: $\mathbf{H}^{\ddagger} = [\mathbf{H}^{H}\mathbf{H}]^{-1}\mathbf{H}^{H}$
- $\mathbf{H}\mathbf{H}^{\ddagger} = \mathbf{I}$



N parallel SISO P2P channels!



Channel Diagonalization

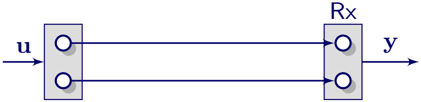
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

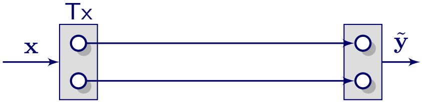
- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



N parallel SISO P2P channels!

$M \leq N$: ZF post-coding:

- Pseudo-inverse: $\mathbf{H}^{\ddagger} = [\mathbf{H}^{H}\mathbf{H}]^{-1}\mathbf{H}^{H}$
- $\mathbf{H}\mathbf{H}^{\ddagger} = \mathbf{I}$



M parallel SISO P2P channels!

Channel Diagonalization

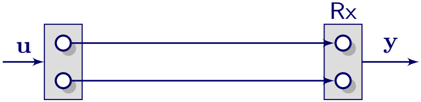
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

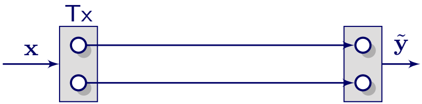
- Pseudo-inverse: $\mathbf{H}^{\dagger} = \mathbf{H}^H [\mathbf{H}\mathbf{H}^H]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



N parallel SISO P2P channels!

$M \leq N$: ZF post-coding:

- Pseudo-inverse: $\mathbf{H}^{\ddagger} = [\mathbf{H}^H \mathbf{H}]^{-1} \mathbf{H}^H$
- $\mathbf{H}\mathbf{H}^{\ddagger} = \mathbf{I}$



M parallel SISO P2P channels!

DoF achievable by treating each sub-channel separately $\Rightarrow$ Separability!

Channel Diagonalization

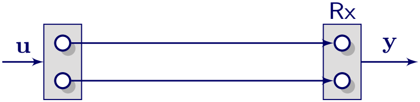
Definition (Channel diagonlization)

Transform an arbitrary MIMO channel matrix $\mathbf{H}$ to a diagonal matrix.

MIMO $M \times N$ P2P channel can be diagonalized by zero-forcing (ZF)

$M \geq N$: ZF pre-coding:

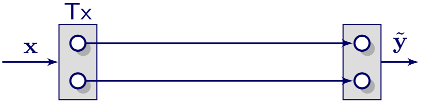
- Pseudo-inverse:
  $\mathbf{H}^{\dagger} = \mathbf{H}^{H}[\mathbf{H}\mathbf{H}^{H}]^{-1}$
- $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{I}$



N parallel SISO P2P channels!

$M \leq N$: ZF post-coding:

- Pseudo-inverse:
  $\mathbf{H}^{\ddagger} = [\mathbf{H}^{H}\mathbf{H}]^{-1}\mathbf{H}^{H}$
- $\mathbf{H}\mathbf{H}^{\ddagger} = \mathbf{I}$



M parallel SISO P2P channels!

DoF achievable by treating each sub-channel separately $\Rightarrow$ Separability!

MAC and BC are also separable

Outline

Signal Alignment

Definition (Signal alignment)

Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that $\text{span}(\mathbf{x}_1) = \text{span}(\mathbf{x}_2)$.

Signal Alignment

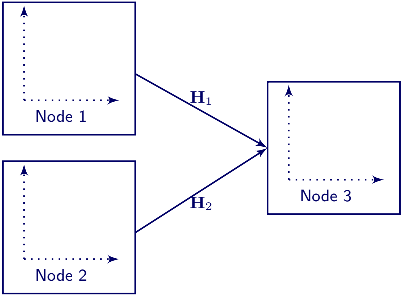
Definition (Signal alignment)

Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that $\text{span}(\mathbf{x}_1) = \text{span}(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$



Signal Alignment

Definition (Signal alignment)

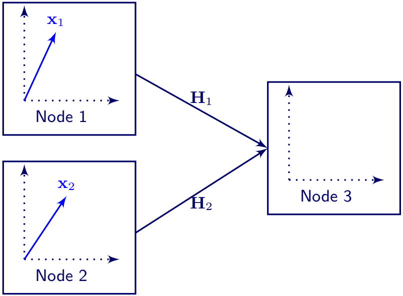
Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that $\text{span}(\mathbf{x}_1) = \text{span}(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$

- $\mathbf{V}_1$, $\mathbf{V}_2$ arbitrary



Signal Alignment

Definition (Signal alignment)

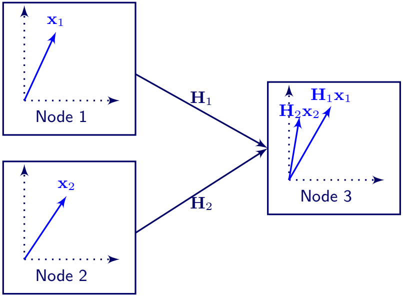
Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that $\text{span}(\mathbf{x}_1) = \text{span}(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$

- $\mathbf{V}_1$, $\mathbf{V}_2$ arbitrary



Signal Alignment

Definition (Signal alignment)

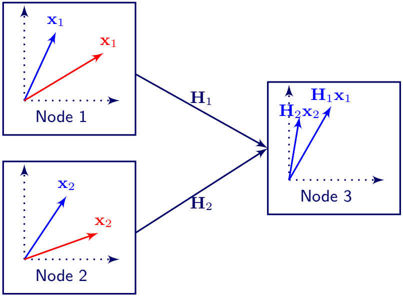
Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that $\text{span}(\mathbf{x}_1) = \text{span}(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$

- $\mathbf{V}_1$, $\mathbf{V}_2$ arbitrary
- $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2$



Signal Alignment

Definition (Signal alignment)

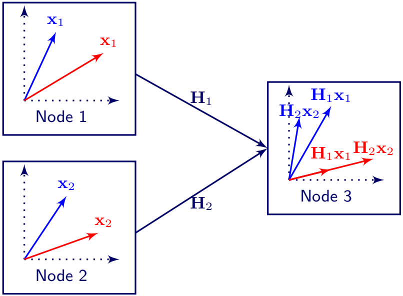
Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that span$(\mathbf{x}_1) =$ span$(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$

- $\mathbf{V}_1$, $\mathbf{V}_2$ arbitrary
- $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2$



Signal Alignment

Definition (Signal alignment)

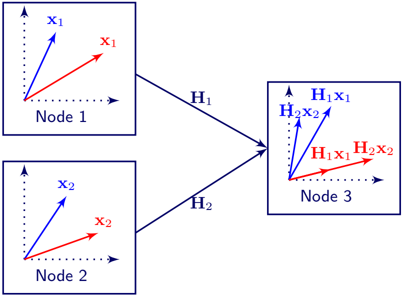
Placing two signals $\mathbf{x}_1$ and $\mathbf{x}_2$ in signal space so that span$(\mathbf{x}_1)$ = span$(\mathbf{x}_2)$.

Two signals can be aligned by pre-coding:

$\mathbf{x}_1 = \mathbf{V}_1 u_1$

$\mathbf{x}_2 = \mathbf{V}_2 u_2$

- $\mathbf{V}_1$, $\mathbf{V}_2$ arbitrary
- $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2$
- Useful for CF!



Signal-alignment for CF

Definition (Compute-forward (CF))

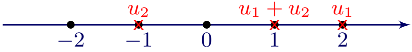
Computing and forwarding a linear combination of two signals e.g. $\mathbf{x}_1 + \mathbf{x}_2$.

Signal-alignment for CF

Definition (Compute-forward (CF))

Computing and forwarding a linear combination of two signals e.g. $\mathbf{x}_1 + \mathbf{x}_2$.

Recall: CF can be accomplished by using lattice codes.

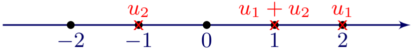


Signal-alignment for CF

Definition (Compute-forward (CF))

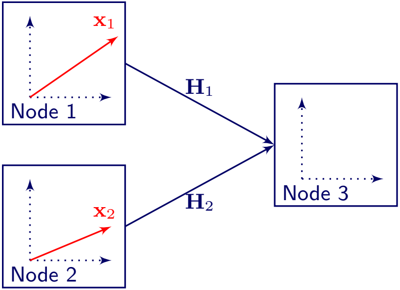
Computing and forwarding a linear combination of two signals e.g. $\mathbf{x}_1 + \mathbf{x}_2$.

Recall: CF can be accomplished by using lattice codes.



Example:

- $\mathbf{x}_1 = \mathbf{V}_1 u_1$,
  $\mathbf{x}_2 = \mathbf{V}_2 u_2$,
  $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2 = [1,\ 1]^T$

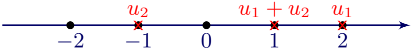


Signal-alignment for CF

Definition (Compute-forward (CF))

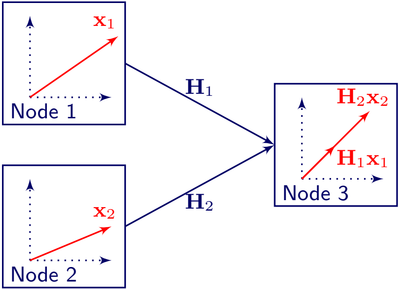
Computing and forwarding a linear combination of two signals e.g. $\mathbf{x}_1 + \mathbf{x}_2$.

Recall: CF can be accomplished by using lattice codes.



Example:

- $\mathbf{x}_1 = \mathbf{V}_1 u_1,$
  $\mathbf{x}_2 = \mathbf{V}_2 u_2,$
  $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2 = [1,\ 1]^T$
- $\mathbf{y}_3 = (u_1 + u_2)[1,\ 1]^T + \mathbf{n}$

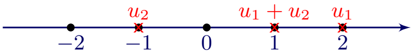


Signal-alignment for CF

Definition (Compute-forward (CF))

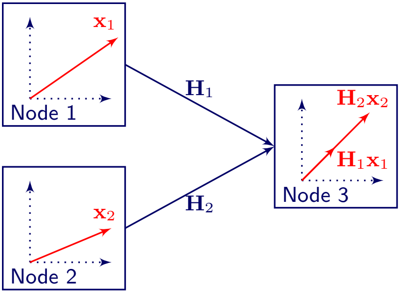
Computing and forwarding a linear combination of two signals e.g. $\mathbf{x}_1 + \mathbf{x}_2$.

Recall: CF can be accomplished by using lattice codes.

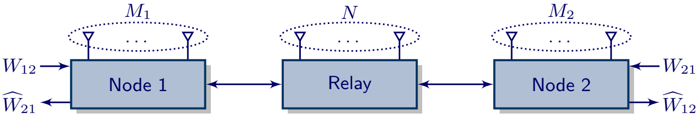


Example:

- $\mathbf{x}_1 = \mathbf{V}_1 u_1,$
  $\mathbf{x}_2 = \mathbf{V}_2 u_2,$
  $\mathbf{H}_1\mathbf{V}_1 = \mathbf{H}_2\mathbf{V}_2 = [1,\ 1]^T$
- $\mathbf{y}_3 = (u_1 + u_2)[1,\ 1]^T + \mathbf{n}$
- compute $u_1 + u_2$ from $[1,\ 1]\mathbf{y}_3 = 2(u_1 + u_2) + n'$

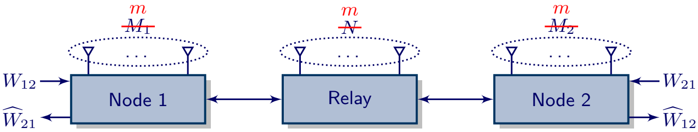


Back to the MIMO Two-way Relay Channel



- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.

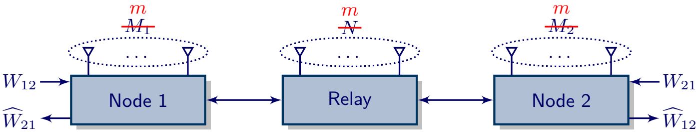
Back to the MIMO Two-way Relay Channel



- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.
- Use only $m = \frac{d}{2} = \min\{M_1, M_2, N\}$ antennas at each transceiver:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 + \mathbf{z}_r, \quad \mathbf{y}_i = \mathbf{D}_i\mathbf{x}_r + \mathbf{z}_i, \quad \mathbf{H}_i, \mathbf{D}_i\colon\ m \times m.$

Back to the MIMO Two-way Relay Channel



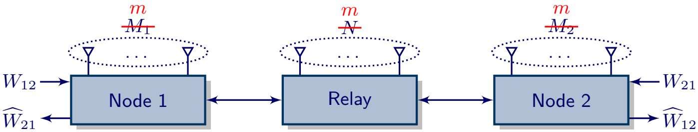
- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.
- Use only $m = \frac{d}{2} = \min\{M_1, M_2, N\}$ antennas at each transceiver:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 + \mathbf{z}_r, \quad \mathbf{y}_i = \mathbf{D}_i\mathbf{x}_r + \mathbf{z}_i, \quad \mathbf{H}_i, \mathbf{D}_i\text{: } m \times m.$

- Channel diagonalization: $\mathbf{x}_i = \mathbf{H}_i^{-1}\mathbf{u}_i$, $\mathbf{y}_i' = \mathbf{D}_i^{-1}\mathbf{y}_i$:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{z}_r, \quad \mathbf{y}_i' = \mathbf{x}_r + \mathbf{z}_i',$

Back to the MIMO Two-way Relay Channel



- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.
- Use only $m = \frac{d}{2} = \min\{M_1, M_2, N\}$ antennas at each transceiver:

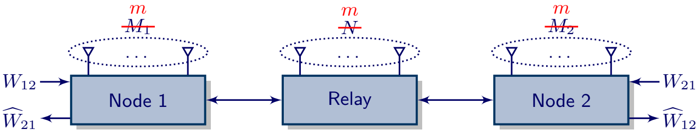
$\Rightarrow$ $\mathbf{y}_r = \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 + \mathbf{z}_r$, $\quad \mathbf{y}_i = \mathbf{D}_i\mathbf{x}_r + \mathbf{z}_i$, $\quad \mathbf{H}_i, \mathbf{D}_i$: $m \times m$.

- Channel diagonalization: $\mathbf{x}_i = \mathbf{H}_i^{-1}\mathbf{u}_i$, $\mathbf{y}_i' = \mathbf{D}_i^{-1}\mathbf{y}_i$:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{z}_r$, $\quad \mathbf{y}_i' = \mathbf{x}_r + \mathbf{z}_i'$,

$\Rightarrow$ m parallel SISO two-way relay channels

Back to the MIMO Two-way Relay Channel



- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.
- Use only $m = \frac{d}{2} = \min\{M_1, M_2, N\}$ antennas at each transceiver:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 + \mathbf{z}_r, \quad \mathbf{y}_i = \mathbf{D}_i\mathbf{x}_r + \mathbf{z}_i, \quad \mathbf{H}_i, \mathbf{D}_i$: $m \times m$.

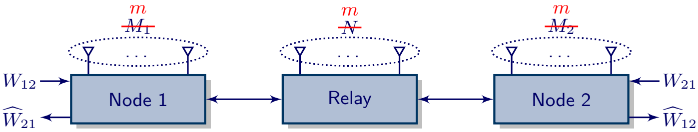
- Channel diagonalization: $\mathbf{x}_i = \mathbf{H}_i^{-1}\mathbf{u}_i$, $\mathbf{y}_i' = \mathbf{D}_i^{-1}\mathbf{y}_i$:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{z}_r, \quad \mathbf{y}_i' = \mathbf{x}_r + \mathbf{z}_i',$

$\Rightarrow$ m parallel SISO two-way relay channels

- Apply CF: achieve $2\left[\frac{1}{2}\log\left(\frac{1}{2} + P\right)\right]^+ \approx 2C(P)$ (at high P) per sub-channel

Back to the MIMO Two-way Relay Channel



- Cut-set bound: $d \leq 2\min\{M_1, M_2, N\}$.
- Use only $m = \frac{d}{2} = \min\{M_1, M_2, N\}$ antennas at each transceiver:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 + \mathbf{z}_r, \quad \mathbf{y}_i = \mathbf{D}_i\mathbf{x}_r + \mathbf{z}_i, \quad \mathbf{H}_i, \mathbf{D}_i\text{: } m \times m.$

- Channel diagonalization: $\mathbf{x}_i = \mathbf{H}_i^{-1}\mathbf{u}_i$, $\mathbf{y}_i' = \mathbf{D}_i^{-1}\mathbf{y}_i$:

$\Rightarrow$ $\mathbf{y}_r = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{z}_r, \quad \mathbf{y}_i' = \mathbf{x}_r + \mathbf{z}_i',$

$\Rightarrow$ m parallel SISO two-way relay channels

- Apply CF: achieve $2\left[\frac{1}{2}\log\left(\frac{1}{2}+P\right)\right]^+ \approx 2C(P)$ (at high P) per sub-channel
- Total rate $2mC(P) \Rightarrow 2m\ (= d)$ DoF

Remarks

- Diagonalization ensures the alignment of each pair of signals in a 1-D space

Remarks

- Diagonalization ensures the alignment of each pair of signals in a 1-D space
- Optimal DoF achievable by using either compress-forward, compute-forward, or amplify-forward over each sub-channel (dimension)

Possible improvement

DoF achieving scheme:

- Reduce the number of antennas to $\min\{M_1, M_2, N\}$,
- apply MIMO pre-coding and post-coding for channel diagonalization,

$\Rightarrow$ decompose channel into sub-channels,

- apply CF over each sub-channel

Possible improvement

DoF achieving scheme:

- Reduce the number of antennas to $\min\{M_1, M_2, N\}$,
- apply MIMO pre-coding and post-coding for channel diagonalization,

$\Rightarrow$ decompose channel into sub-channels,

- apply CF over each sub-channel

Use excess antennas for sending a (uni-directional) DF signal Dirty-paper coded against the remaining signals at the same node.

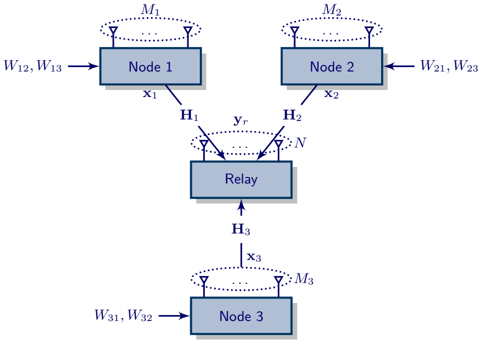
Outline

1. From Capacity to DoF
2. MIMO Two-Way Relay Channel
   - Channel diagonalization
   - Signal Alignment
3. MIMO multi-way relay channel
   - Sum-DoF
   - DoF Region
4. MIMO Multi-way Channel

MIMO Multi-way Relay Channel

MIMO Y-channel [Lee & Lim 09]

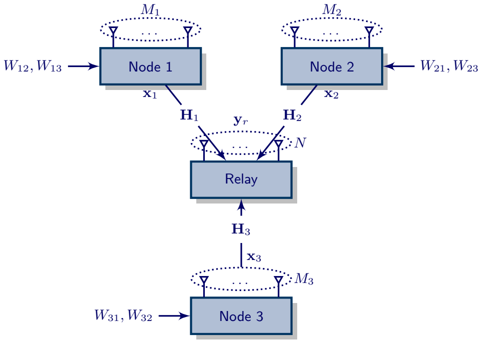
- inputs $\mathbf{x}_i$ and $\mathbf{x}_r$,
- outputs $\mathbf{y}_r = \sum_{i=1}^{K} \mathbf{H}_i \mathbf{x}_i + \mathbf{z}_r$ and $\mathbf{y}_i = \mathbf{D}_i \mathbf{x}_r + \mathbf{z}_i$,



MIMO Multi-way Relay Channel

MIMO Y-channel [Lee & Lim 09]

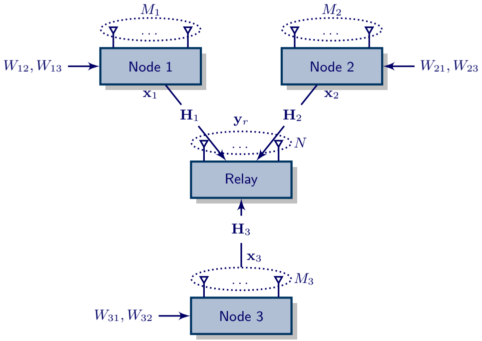
- inputs $\mathbf{x}_i$ and $\mathbf{x}_r$,
- outputs $\mathbf{y}_r = \sum_{i=1}^{K} \mathbf{H}_i \mathbf{x}_i + \mathbf{z}_r$ and $\mathbf{y}_i = \mathbf{D}_i \mathbf{x}_r + \mathbf{z}_i$,
- channels $\mathbf{H}_i$: $N \times M_i$, and $\mathbf{D}_i$: $M_i \times N$,



MIMO Multi-way Relay Channel

MIMO Y-channel [Lee & Lim 09]

- inputs $\mathbf{x}_i$ and $\mathbf{x}_r$,
- outputs $\mathbf{y}_r = \sum_{i=1}^{K} \mathbf{H}_i \mathbf{x}_i + \mathbf{z}_r$ and $\mathbf{y}_i = \mathbf{D}_i \mathbf{x}_r + \mathbf{z}_i$,
- channels $\mathbf{H}_i$: $N \times M_i$, and $\mathbf{D}_i$: $M_i \times N$,
- Question: DoF?



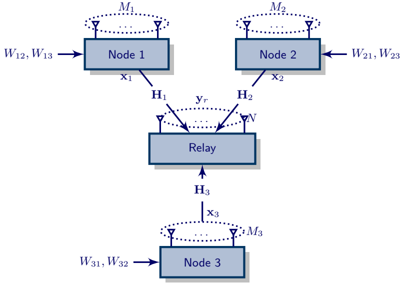
Outline

1. From Capacity to DoF
2. MIMO Two-Way Relay Channel
   - Channel diagonalization
   - Signal Alignment
3. MIMO multi-way relay channel
   - Sum-DoF
   - DoF Region
4. MIMO Multi-way Channel

MIMO Multi-way Relay Channel

- If $M_1 = M_2 = M_3 = M$ and $N \geq \lceil 3M/2 \rceil$, then:
- cut-set bound is achievable [Lee *et al.* 10],
- Achievability: Signal-space alignment for NC

$\Rightarrow$ i.e., $d = 3M$,

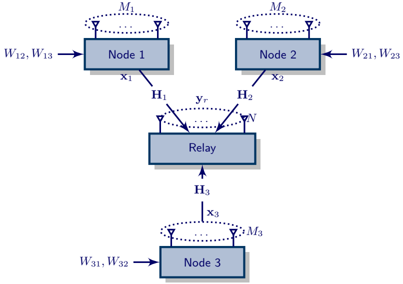


MIMO Multi-way Relay Channel

- If $M_1 = M_2 = M_3 = M$ and $N \geq \lceil 3M/2 \rceil$, then:
- cut-set bound is achievable [Lee *et al.* 10],
- Achievability: Signal-space alignment for NC

$\Rightarrow$ i.e., $d = 3M$,

- Question: What if $N \leq \lceil 3M/2 \rceil$?

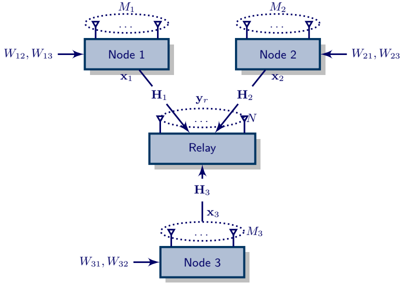


MIMO Multi-way Relay Channel

- If $M_1 = M_2 = M_3 = M$ and $N \geq \lceil 3M/2 \rceil$, then:
- cut-set bound is achievable [Lee *et al.* 10],
- Achievability: Signal-space alignment for NC

$\Rightarrow$ i.e., $d = 3M$,

- Question: What if $N \leq \lceil 3M/2 \rceil$?



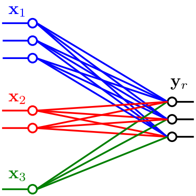
sum-DoF

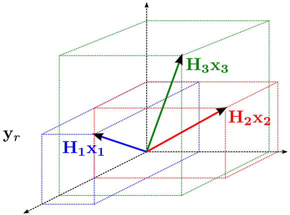
The sum-DoF of the MIMO Y-channel with $M_1 \geq M_2 \geq M_3$ (wlog) is given by

$$d = \min\{\underbrace{M_1 + M_2 + M_3}_{\text{Cut-set bound}}, \underbrace{2M_2 + 2M_3, 2N}_{\text{New bounds}}\}.$$

Transmission Strategy

Signal-space alignment for network-coding

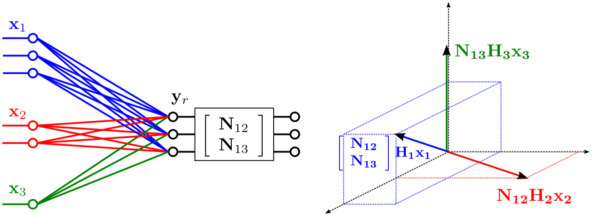




- The signals $\mathbf{H_1x_1}$, $\mathbf{H_2x_2}$, and $\mathbf{H_3x_3}$ fill the entire space at the relay

Transmission Strategy

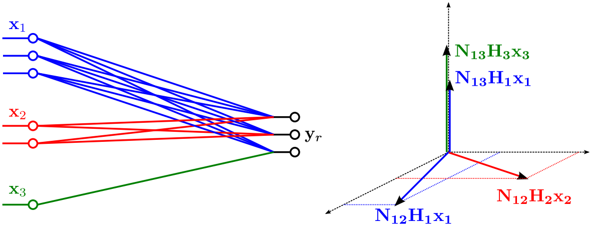
Signal-space alignment for network-coding



- The signals $\mathbf{H_1x_1}$, $\mathbf{H_2x_2}$, and $\mathbf{H_3x_3}$ fill the entire space at the relay
- Relay zero-forces $\mathbf{H_2x_2}$ using $\mathbf{N_{13}}$ and $\mathbf{H_3x_3}$ using $\mathbf{N_{12}}$

Transmission Strategy

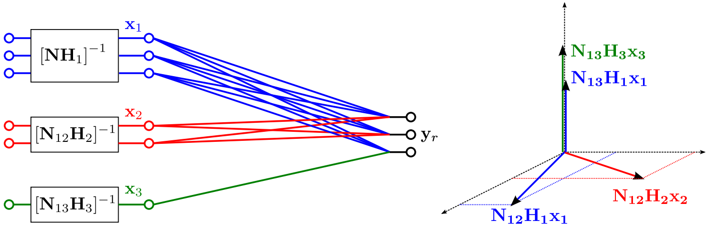
Signal-space alignment for network-coding



- The signals $\mathbf{H_1x_1}$, $\mathbf{H_2x_2}$, and $\mathbf{H_3x_3}$ fill the entire space at the relay
- Relay zero-forces $\mathbf{H_2x_2}$ using $\mathbf{N_{13}}$ and $\mathbf{H_3x_3}$ using $\mathbf{N_{12}}$
- Result: $\mathbf{N_{12}H_1x_1} + \mathbf{N_{12}H_2x_2}$ (2D) and $\mathbf{N_{13}H_1x_1} + \mathbf{N_{13}H_3x_3}$ (1D)

Transmission Strategy

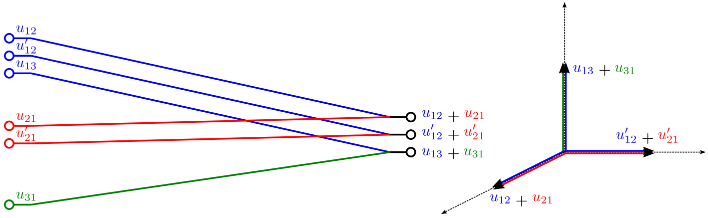
Signal-space alignment for network-coding



- The signals $\mathbf{H_1x_1}$, $\mathbf{H_2x_2}$, and $\mathbf{H_3x_3}$ fill the entire space at the relay
- Relay zero-forces $\mathbf{H_2x_2}$ using $\mathbf{N_{13}}$ and $\mathbf{H_3x_3}$ using $\mathbf{N_{12}}$
- Result: $\mathbf{N_{12}H_1x_1} + \mathbf{N_{12}H_2x_2}$ (2D) and $\mathbf{N_{13}H_1x_1} + \mathbf{N_{13}H_3x_3}$ (1D)
- Tx's diagonalize their effective channels

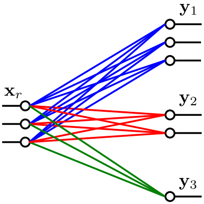
Transmission Strategy

Signal-space alignment for network-coding

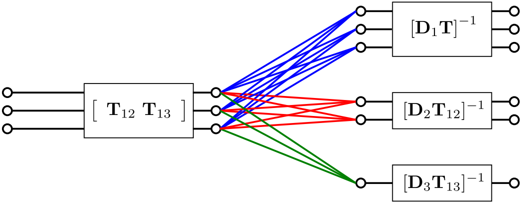


- The signals $\mathbf{H_1x_1}$, $\mathbf{H_2x_2}$, and $\mathbf{H_3x_3}$ fill the entire space at the relay
- Relay zero-forces $\mathbf{H_2x_2}$ using $\mathbf{N_{13}}$ and $\mathbf{H_3x_3}$ using $\mathbf{N_{12}}$
- Result: $\mathbf{N_{12}H_1x_1}+\mathbf{N_{12}H_2x_2}$ (2D) and $\mathbf{N_{13}H_1x_1}+\mathbf{N_{13}H_3x_3}$ (1D)
- Tx's diagonalize their effective channels $\Rightarrow$ desired channel structure
- Relay obtains net-coded signals: $u_{12}+u_{21}$, $u'_{12}+u'_{21}$, and $u_{13}+u_{31}$

Transmission strategy

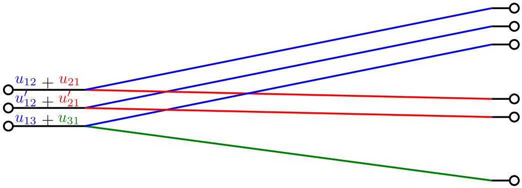


Transmission strategy



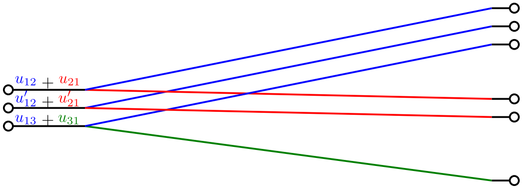
- Relay uses a similar beam-forming strategy to deliver the net-coded signals to their desired destinations

Transmission strategy



- Relay uses a similar beam-forming strategy to deliver the net-coded signals to their desired destinations
- User 1 gets $u_{12} + u_{21}$, $u'_{12} + u'_{21}$, and $u_{13} + u_{31}$, and extracts u_{21}, u'_{21}, and u_{31}
- User 2 gets $u_{12} + u_{21}$ and $u'_{12} + u'_{21}$ and extracts u_{12} and u'_{12}
- User 3 gets $u_{13} + u_{31}$ and extracts u_{13}

Transmission strategy



- Relay uses a similar beam-forming strategy to deliver the net-coded signals to their desired destinations
- User 1 gets $u_{12} + u_{21}$, $u'_{12} + u'_{21}$, and $u_{13} + u_{31}$, and extracts u_{21}, u'_{21}, and u_{31}
- User 2 gets $u_{12} + u_{21}$ and $u'_{12} + u'_{21}$ and extracts u_{12} and u'_{12}
- User 3 gets $u_{13} + u_{31}$ and extracts u_{13}
- 6 symbols delivered successfully $\Rightarrow$ 6 DoF (optimal)

General Transmission Strategy

Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

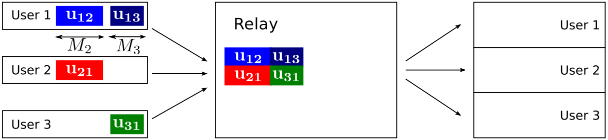
- Use only $M_2 + M_3$ antennas at the relay

General Transmission Strategy

Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

- Use only $M_2 + M_3$ antennas at the relay
- Users send $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ (M_2-dim), and $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ (M_3-dim)

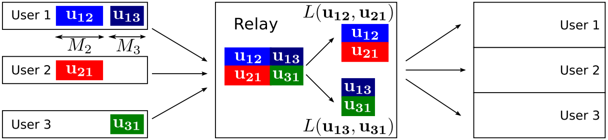
General Transmission Strategy



Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

- Use only $M_2 + M_3$ antennas at the relay
- Users send $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ (M_2-dim), and $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ (M_3-dim)
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$, and align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ @ relay

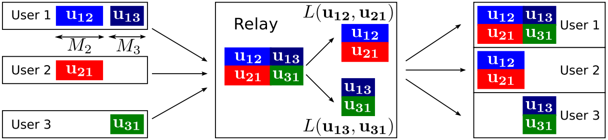
General Transmission Strategy



Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

- Use only $M_2 + M_3$ antennas at the relay
- Users send $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ (M_2-dim), and $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ (M_3-dim)
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$, and align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ @ relay
- Relay decodes $L(\mathbf{u}_{12},\mathbf{u}_{21})$ and $L(\mathbf{u}_{13},\mathbf{u}_{31})$

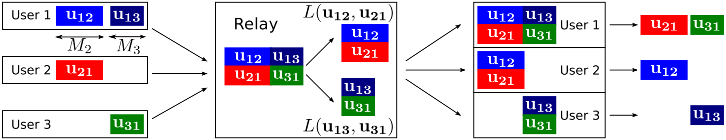
General Transmission Strategy



Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

- Use only $M_2 + M_3$ antennas at the relay
- Users send $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ (M_2-dim), and $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ (M_3-dim)
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$, and align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ @ relay
- Relay decodes $L(\mathbf{u}_{12}, \mathbf{u}_{21})$ and $L(\mathbf{u}_{13}, \mathbf{u}_{31})$
- Beam-form $L(\mathbf{u}_{12}, \mathbf{u}_{21})$ and $L(\mathbf{u}_{13}, \mathbf{u}_{31})$ orthogonal to users 3 and 2, respectively

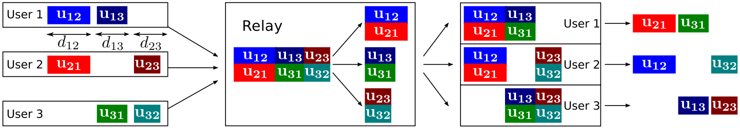
General Transmission Strategy



Case 1: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2M_2 + 2M_3$

- Use only $M_2 + M_3$ antennas at the relay
- Users send $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ (M_2-dim), and $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ (M_3-dim)
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$, and align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ @ relay
- Relay decodes $L(\mathbf{u}_{12}, \mathbf{u}_{21})$ and $L(\mathbf{u}_{13}, \mathbf{u}_{31})$
- Beam-form $L(\mathbf{u}_{12}, \mathbf{u}_{21})$ and $L(\mathbf{u}_{13}, \mathbf{u}_{31})$ orthogonal to users 3 and 2, respectively
- Each user decodes the desired linear combinations, and extracts the desired signals $\Rightarrow 2M_2 + 2M_3$ DoF

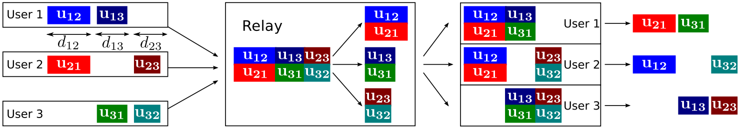
General Transmission Strategy



Case 2: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = M_1 + M_2 + M_3$

- Similar to [Lee *et al.* 10] but with an asymmetric DoF allocation

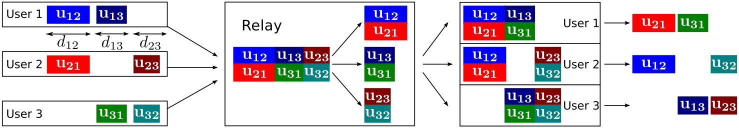
General Transmission Strategy



Case 2: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = M_1 + M_2 + M_3$

- Similar to [Lee *et al.* 10] but with an asymmetric DoF allocation
- Use only $\frac{M_1+M_2+M_3}{2}$ antennas at the relay

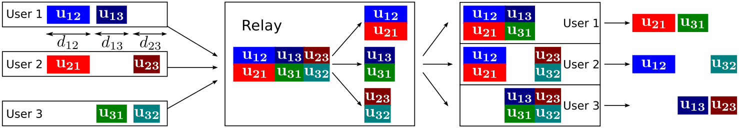
General Transmission Strategy



Case 2: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = M_1 + M_2 + M_3$

- Similar to [Lee *et al.* 10] but with an asymmetric DoF allocation
- Use only $\frac{M_1+M_2+M_3}{2}$ antennas at the relay
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ in a d_{12}-dim subspace ($d_{12} = \dim(\text{span}(\mathbf{H}_1) \cap \text{span}(\mathbf{H}_2)) = \frac{M_1+M_2-M_3}{2}$)

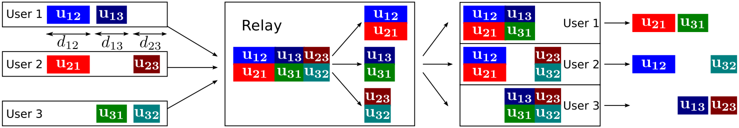
General Transmission Strategy



Case 2: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = M_1 + M_2 + M_3$

- Similar to [Lee *et al.* 10] but with an asymmetric DoF allocation
- Use only $\frac{M_1+M_2+M_3}{2}$ antennas at the relay
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ in a d_{12}-dim subspace ($d_{12} = \text{dim}(\text{span}(\mathbf{H}_1) \cap \text{span}(\mathbf{H}_2)) = \frac{M_1+M_2-M_3}{2}$)
- similarly, align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ in $d_{13} = \frac{M_1+M_3-M_2}{2}$ dimensions, and align $\mathbf{u}_{23}$ and $\mathbf{u}_{32}$ in $d_{23} = \frac{M_2+M_3-M_1}{2}$ dimensions

General Transmission Strategy



Case 2: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = M_1 + M_2 + M_3$

- Similar to [Lee *et al.* 10] but with an asymmetric DoF allocation
- Use only $\frac{M_1+M_2+M_3}{2}$ antennas at the relay
- Align $\mathbf{u}_{12}$ and $\mathbf{u}_{21}$ in a d_{12}-dim subspace ($d_{12} = \text{dim}(\text{span}(\mathbf{H}_1) \cap \text{span}(\mathbf{H}_2)) = \frac{M_1+M_2-M_3}{2}$)
- similarly, align $\mathbf{u}_{13}$ and $\mathbf{u}_{31}$ in $d_{13} = \frac{M_1+M_3-M_2}{2}$ dimensions, and align $\mathbf{u}_{23}$ and $\mathbf{u}_{32}$ in $d_{23} = \frac{M_2+M_3-M_1}{2}$ dimensions
- Achieve $d = 2d_{12} + 2d_{13} + 2d_{23} = M_1 + M_2 + M_3$ DoF

General Transmission Strategy

Case 3: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2N$

- Reduce the number of antennas at the users so that $N = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3\}$

General Transmission Strategy

Case 3: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2N$

- Reduce the number of antennas at the users so that $N = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3\}$
- Use same scheme as **case 1** or **case 2**

General Transmission Strategy

Case 3: $d = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3, 2N\} = 2N$

- Reduce the number of antennas at the users so that $N = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3\}$
- Use same scheme as **case 1** or **case 2**

Remark: If $M_3 = 0 \Rightarrow$ sum-DoF of the two-way relay channel

Outline

1. From Capacity to DoF
2. MIMO Two-Way Relay Channel
   - Channel diagonalization
   - Signal Alignment
3. MIMO multi-way relay channel
   - Sum-DoF
   - DoF Region
4. MIMO Multi-way Channel

Importance of DoF region

Definition

R_{ij}: Rate of signal from i to j

Importance of DoF region

Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

Importance of DoF region

Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_{Σ}: sum-DoF $= \sum d_{ij}$

Importance of DoF region

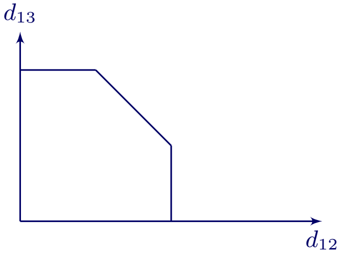
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_Σ: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



Importance of DoF region

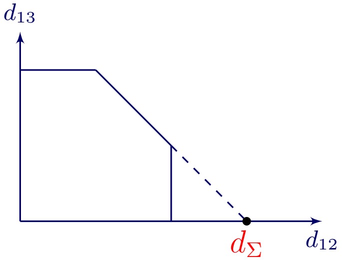
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_Σ: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



+ d_Σ is an overall measure of performance for a network

Importance of DoF region

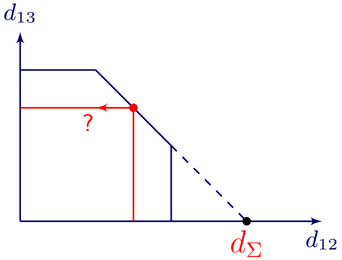
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_{Σ}: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



+ d_{Σ} is an overall measure of performance for a network
− d_{Σ} doesn't provide insights on the trade-off between individual DoF's

Importance of DoF region

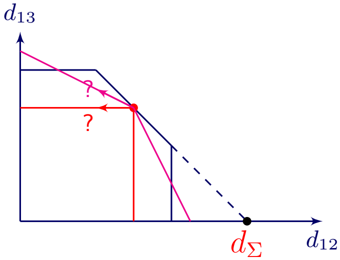
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_Σ: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



+ d_Σ is an overall measure of performance for a network

− d_Σ doesn't provide insights on the trade-off between individual DoF's

Importance of DoF region

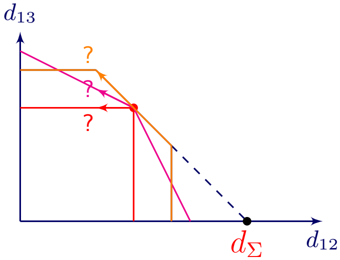
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_{Σ}: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



+ d_{Σ} is an overall measure of performance for a network
− d_{Σ} doesn't provide insights on the trade-off between individual DoF's

Importance of DoF region

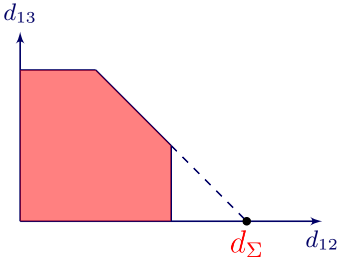
Definition

R_{ij}: Rate of signal from i to j

d_{ij}: DoF defined as $\lim_{P\to\infty} \frac{R_{ij}}{C(P)}$

d_Σ: sum-DoF $= \sum d_{ij}$

$\mathcal{D}$: Set of simultaneously achievable DoF's d_{ij}



+ d_Σ is an overall measure of performance for a network
− d_Σ doesn't provide insights on the trade-off between individual DoF's

Goal

Find the DoF region of the MIMO Y-channel.

DoF Region

- Bi-directional communication suffices for sum-DoF,

DoF Region

- Bi-directional communication suffices for sum-DoF,
- True for DoF-region?

DoF Region

- Bi-directional communication suffices for sum-DoF,
- True for DoF-region?
- No!

Optimal scheme is a combination of bi-directional, cyclic, and uni-directional schemes.

Simple example

Goal: Achieve the DoF tuple: over a Y-channel with $M = N = 3$

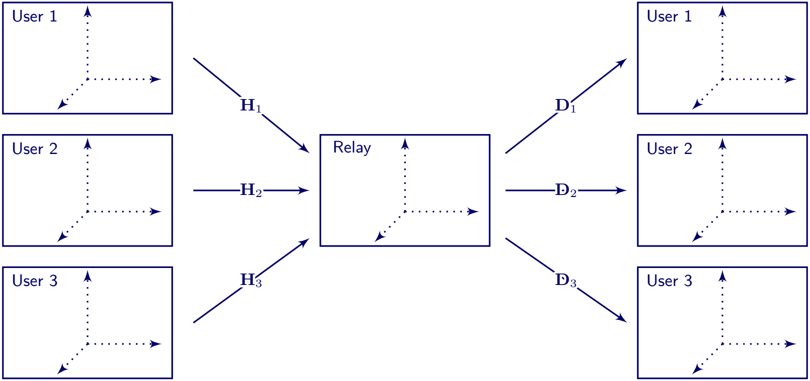
d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
2	0	1	1	1	0

Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
2	0	1	1	1	0

Lets try uni-directional

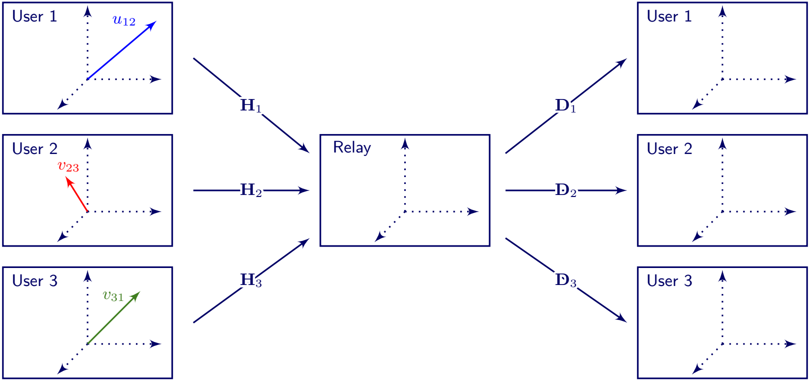


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
2	0	1	1	1	0

Lets try uni-directional

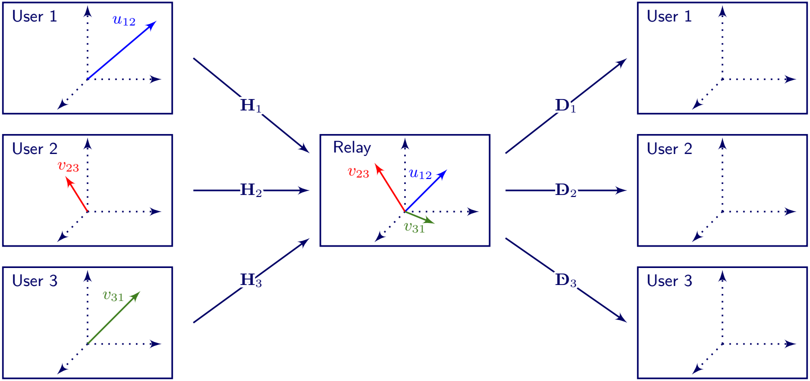


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
2	0	1	1	1	0

Lets try uni-directional

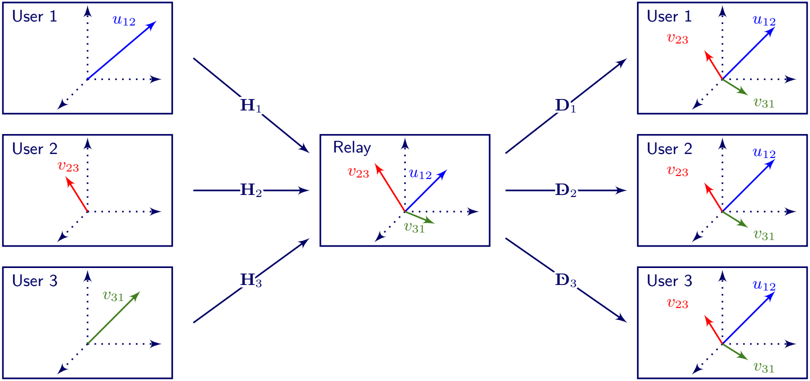


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
2	0	1	1	1	0

Lets try uni-directional

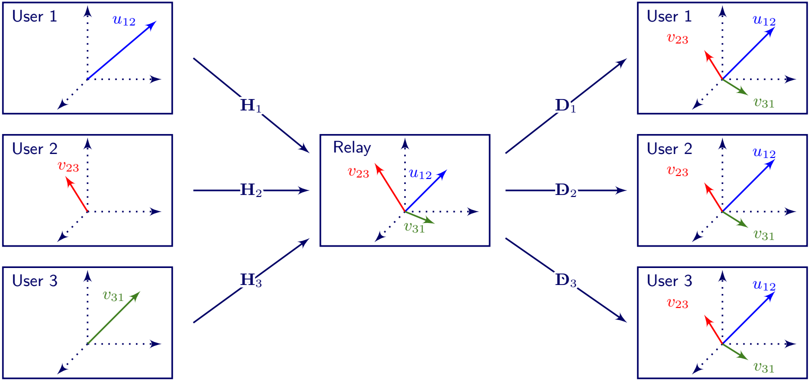


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	1	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional

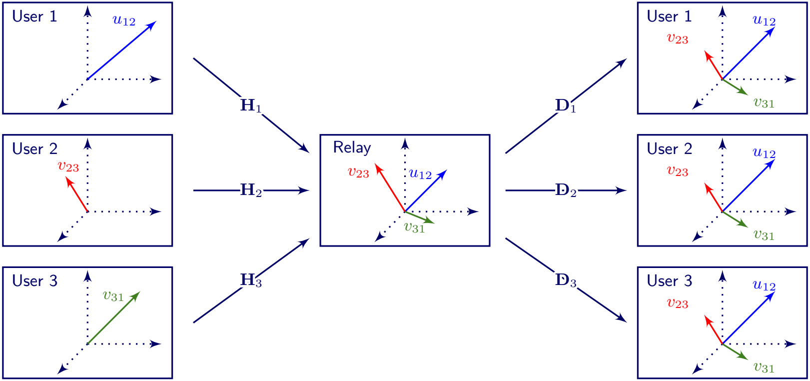


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	1	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional

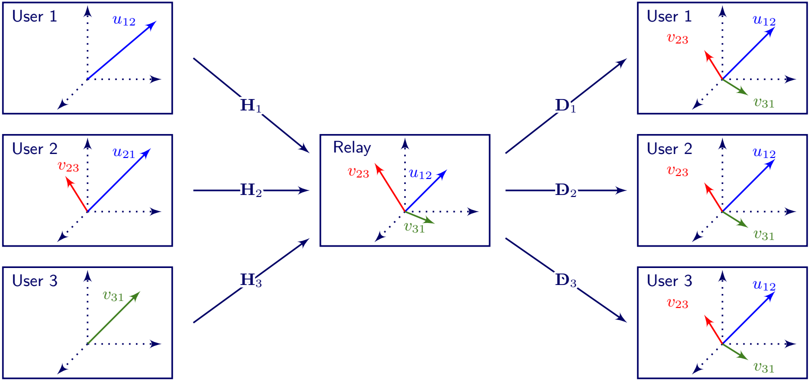


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	1	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional + bi-directional

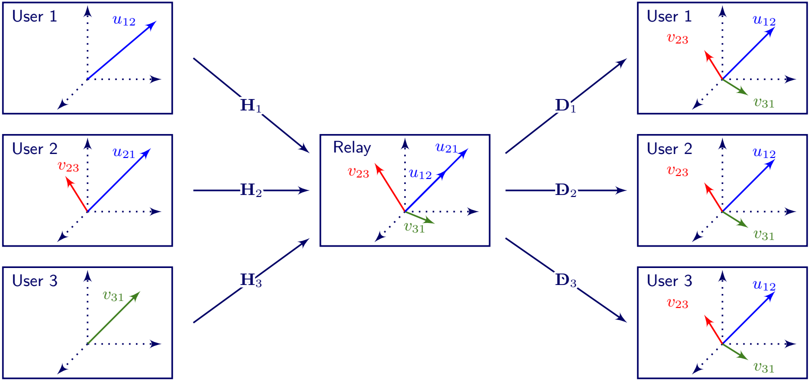


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	1	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional

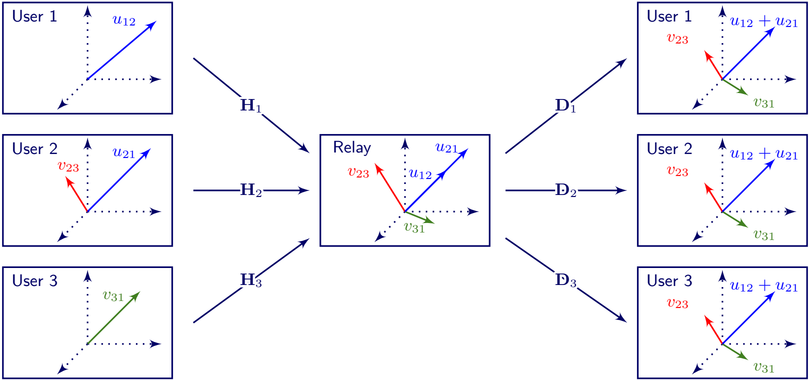


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	1	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional

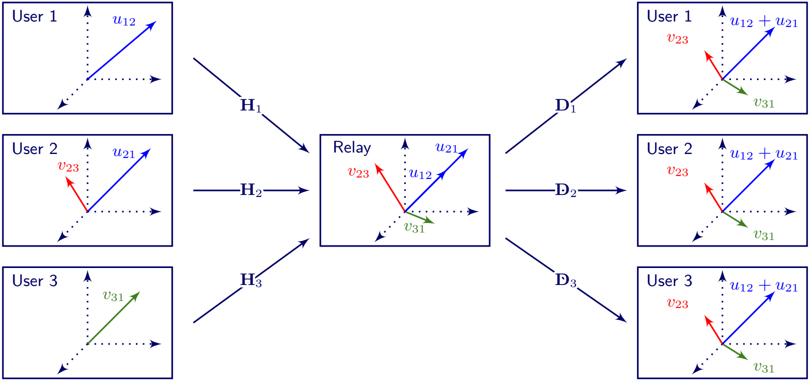


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional

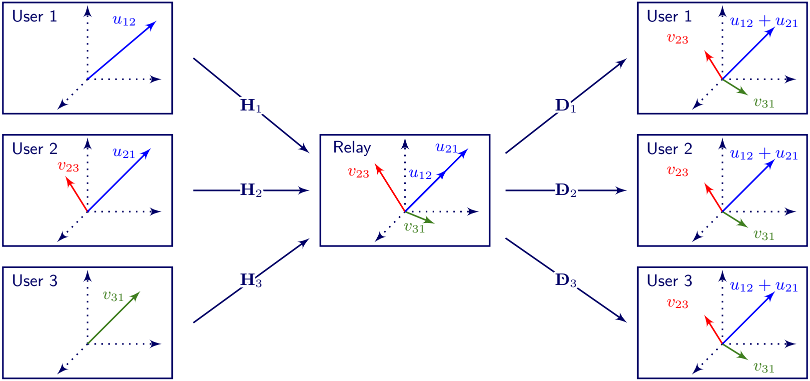


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional+ Cyclic

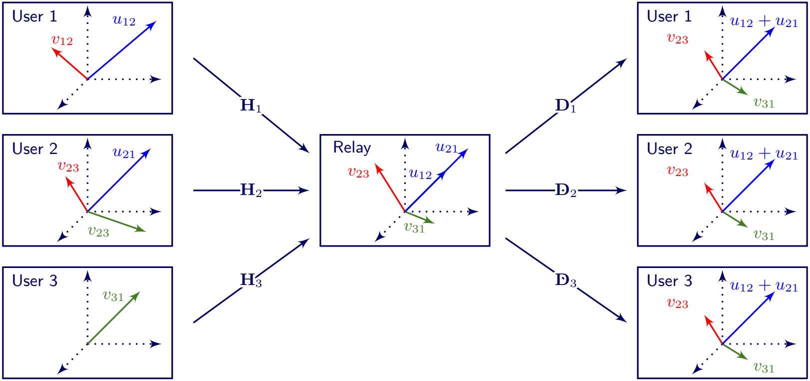


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional+ Cyclic

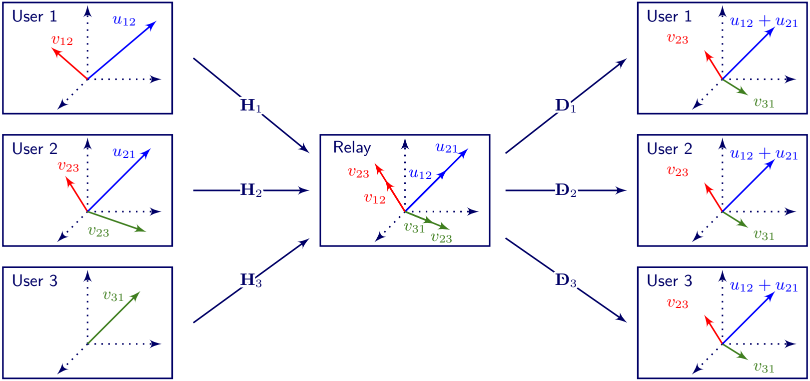


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional+ Cyclic

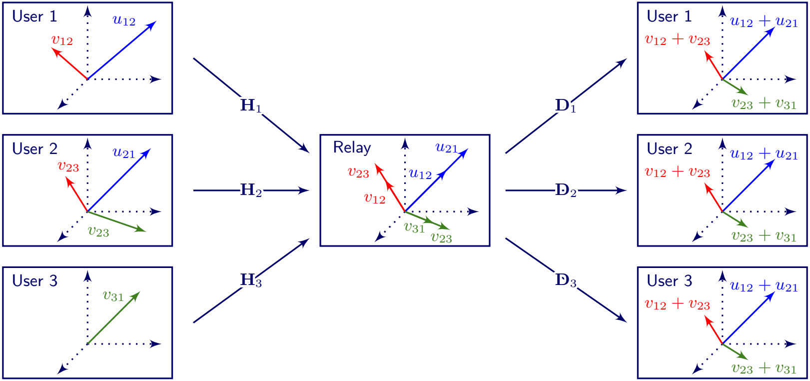


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ 1	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional+ Cyclic

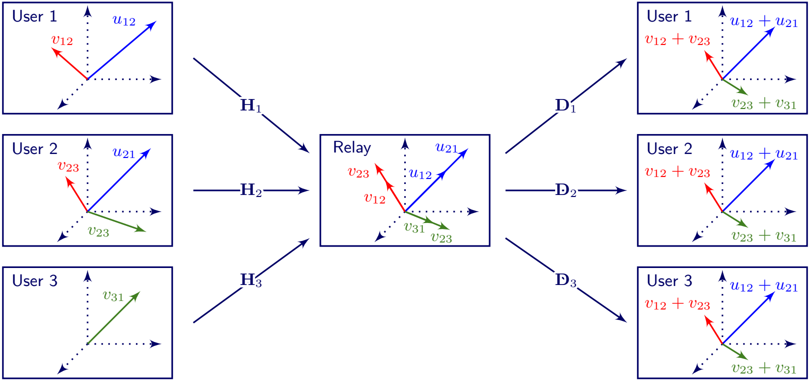


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ ~~1~~ 0	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional+ bi-directional+ Cyclic

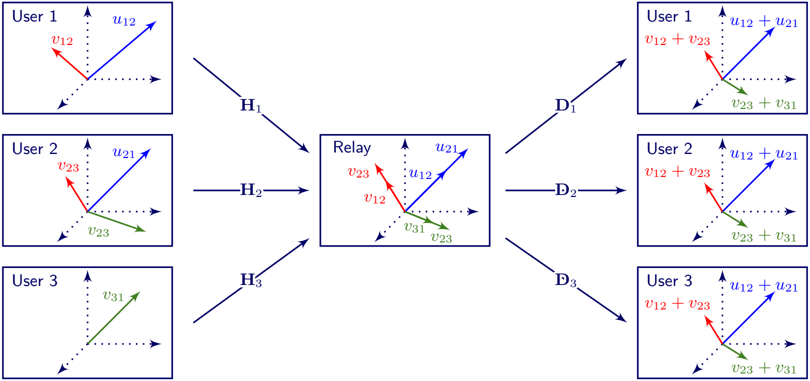


Simple example

Goal: Achieve the DoF tuple:
over a Y-channel with
$M = N = 3$

d_{12}	d_{13}	d_{21}	d_{23}	d_{31}	d_{32}
~~2~~ ~~1~~ 0	0	~~1~~ 0	~~1~~ 0	~~1~~ 0	0

Lets try uni-directional + bi-directional + Cyclic $\leftarrow$ optimal combination



DoF Region

The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$d_{12} + d_{13} + d_{23} \leq N$$

DoF Region

The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$d_{12} + d_{13} + d_{23} \leq N$$
$$d_{12} + d_{13} + d_{32} \leq N$$

DoF Region

The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$\begin{aligned} d_{12} + d_{13} + d_{23} &\leq N \\ d_{12} + d_{13} + d_{32} &\leq N \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad &\leq \vdots \end{aligned}$$

DoF Region

The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$\begin{aligned} d_{12} + d_{13} + d_{23} &\leq N \\ d_{12} + d_{13} + d_{32} &\leq N \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad &\leq \vdots \end{aligned}$$

DoF region [C. & S. 2014]

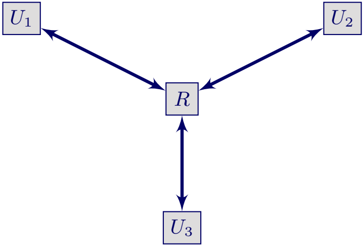
DoF region for $N \leq M$ described by

$$d_{p_1p_2} + d_{p_1p_3} + d_{p_2p_3} \leq N, \quad \forall \mathbf{p}$$

where $\mathbf{p}$ is a permutation of $(1, 2, 3)$ and p_i is its i-th component.

Overview

Achievability of $\mathcal{D}$ is proved using:

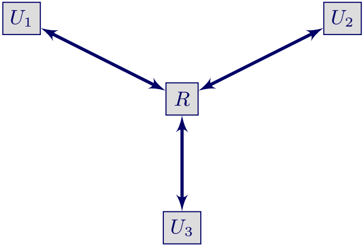


Overview

Achievability of $\mathcal{D}$ is proved using:

Channel diagonalization:

MIMO
Y-channel

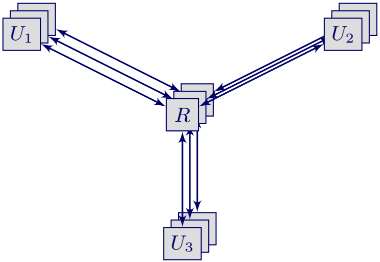


Overview

Achievability of $\mathcal{D}$ is proved using:

Channel diagonalization:

MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)

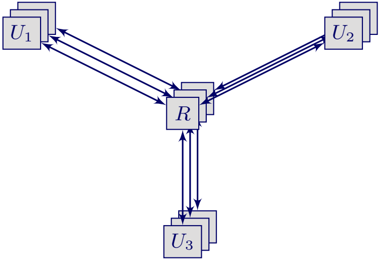


Overview

Achievability of $\mathcal{D}$ is proved using:

Channel diagonalization:

MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)



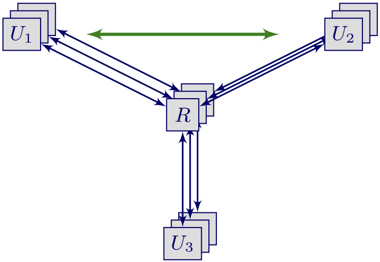
Information exchange:

Overview

Achievability of $\mathcal{D}$ is proved using:

Channel diagonalization:

MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)

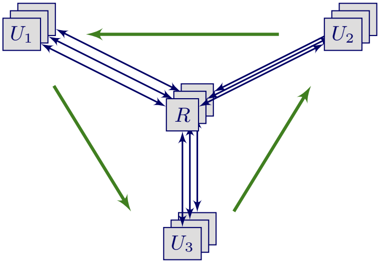


Information exchange:

- Bi-directional: signal-alignment/compute-forward

Overview

Achievability of $\mathcal{D}$ is proved using:



Channel diagonalization:

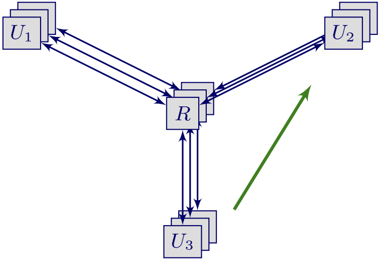
MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)

Information exchange:

- Bi-directional: signal-alignment/compute-forward
- Cyclic: signal-alignment/compute-forward

Overview

Achievability of $\mathcal{D}$ is proved using:



Channel diagonalization:

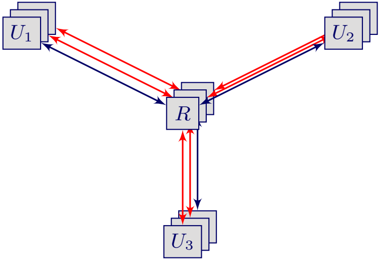
MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)

Information exchange:

- Bi-directional: signal-alignment/compute-forward
- Cyclic: signal-alignment/compute-forward
- Uni-directional: decode-forward

Overview

Achievability of $\mathcal{D}$ is proved using:



Channel diagonalization:

MIMO Y-channel $\rightarrow$ N SISO Y-channels (sub-channels)

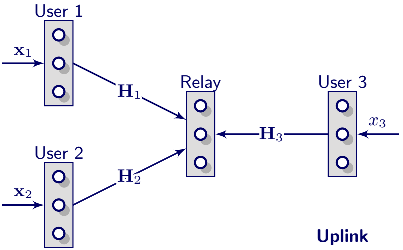
Information exchange:

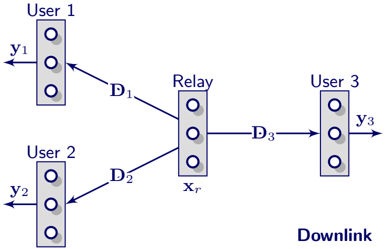
- Bi-directional: signal-alignment/compute-forward
- Cyclic: signal-alignment/compute-forward
- Uni-directional: decode-forward

Resource allocation: distribute sub-channels over users

Channel Diagonalization

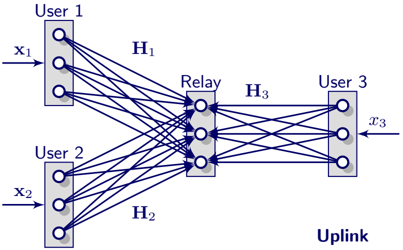
- a MIMO Y-channel with $M = N = 3$

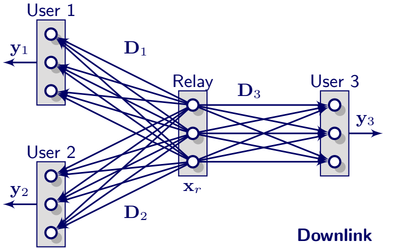




Channel Diagonalization

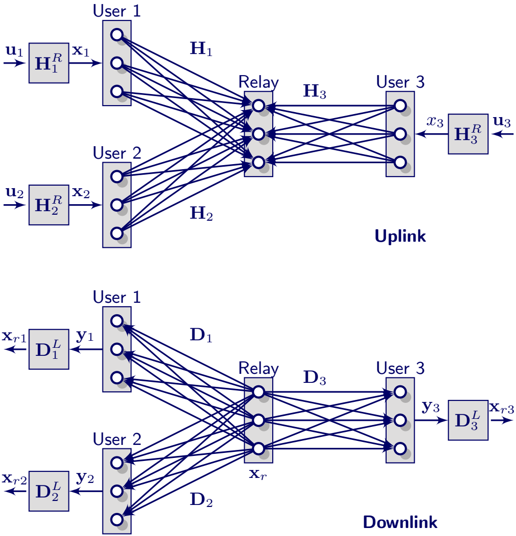
- a MIMO Y-channel with $M = N = 3$
- actually looks like this!





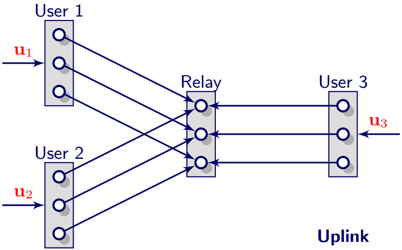
Channel Diagonalization

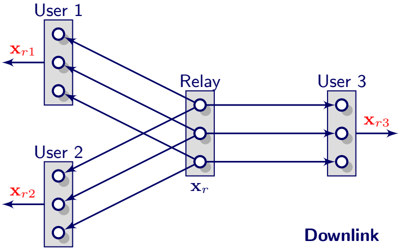
- a MIMO Y-channel with $M = N = 3$
- actually looks like this!
- Pre- and post-code using the Moore-Penrose pseudo inverse



Channel Diagonalization

- a MIMO Y-channel with $M = N = 3$
- actually looks like this!
- Pre- and post-code using the Moore-Penrose pseudo inverse
- Channel Diagonalization $\Rightarrow N$ sub-channels

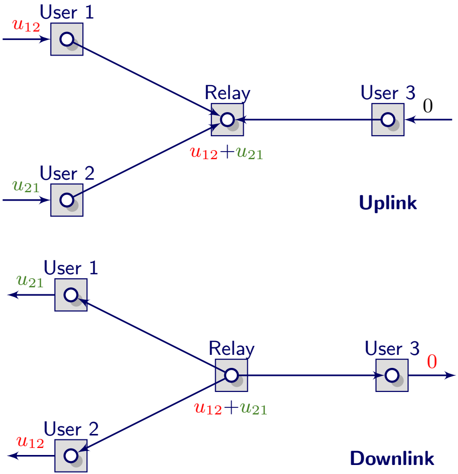




Information transfer

Bi-directional:

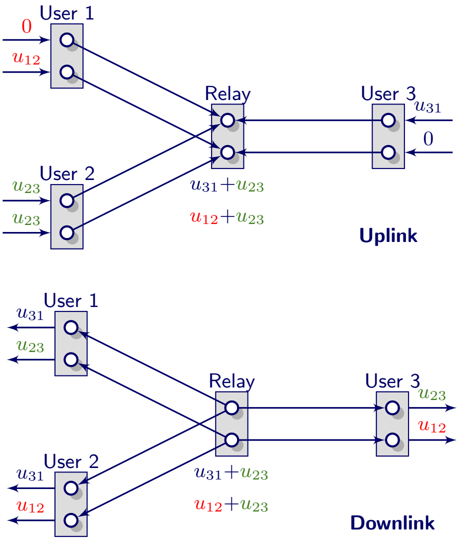
- signal-alignment
- compute-forward
- exchanges 2 symbols
- requires 1 sub-channel (up- and down-link)
- efficiency 2 DoF/dimension



Information transfer

Cyclic:

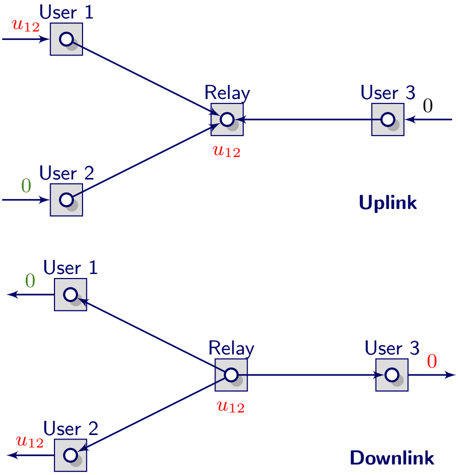
- signal-alignment
- compute-forward
- exchanges 3 symbols
- requires 2 sub-channels (up- and down-link)
- efficiency 3/2 DoF/dimension



Information transfer

Uni-directional:

- decode-forward
- exchanges 1 symbols
- requires 1 sub-channel (up- and down-link)
- efficiency 1 DoF/dimension



Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Uni-directional only:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- cyclic achieves $d_{12}^c = d_{23}^c = d_{31}^c = 1$ over 2 sub-channels

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
- cyclic achieves $d_{12}^c = d_{23}^c = d_{31}^c = 1$ over 2 sub-channels
- residual DoF $(0, 0, 0, 0, 0, 0)$

Back to our example

DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2,0,1,1,1,0)$, Y-channel with $3 = N \leq M$

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Uni-directional only:

- 5 sub-channels $> N$!

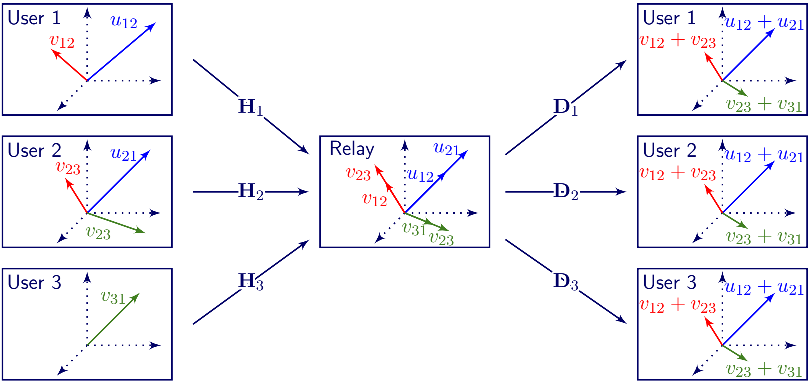
Bi-directional + uni-directional:

- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1,0,0,1,1,0)$
- uni-directional needs 3 more sub-channels
- total number of sub-channels 4 $> N$!

Bi-directional + cyclic + uni-directional:

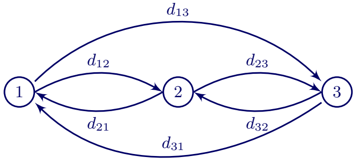
- bi-directional achieves $d_{12}^b = d_{21}^b = 1$ over 1 sub-channel
- residual DoF $(1,0,0,1,1,0)$
- cyclic achieves $d_{12}^c = d_{23}^c = d_{31}^c = 1$ over 2 sub-channels
- residual DoF $(0,0,0,0,0,0)$
- total number of sub-channels 3 $= N$!

Back to our example



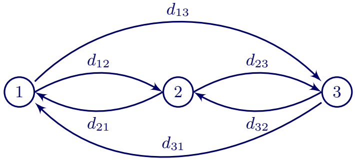
In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$



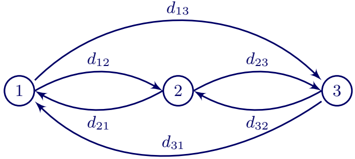
In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!



In General

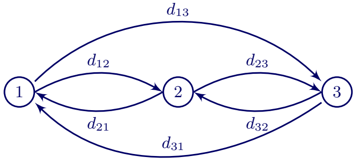
Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ ⇒ 2-cycles and 3-cycles!



Bi-directional:

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

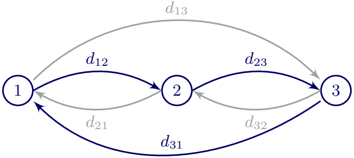


Bi-directional:
resolves 2-cycles

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:
resolves 2-cycles

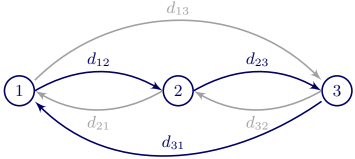


Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ $\Rightarrow$ 3-cycle!

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:
resolves 2-cycles

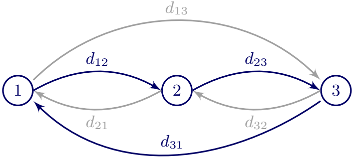


Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!



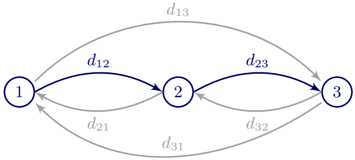
Bi-directional:
resolves 2-cycles

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:
resolves 3-cycles

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!



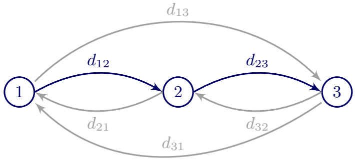
Bi-directional:
resolves 2-cycles

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ $\Rightarrow$ 3-cycle!

Cyclic:
resolves 3-cycles
Residual DoF tuple (e.g.) $\mathbf{d}' = (d''_{12}, 0, 0, d''_{23}, 0, 0)$ $\Rightarrow$ no cycles!

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!



Bi-directional:
resolves 2-cycles

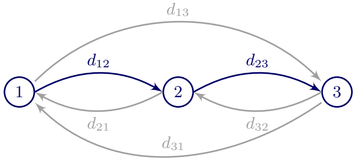
Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:
resolves 3-cycles
Residual DoF tuple (e.g.) $\mathbf{d}' = (d''_{12}, 0, 0, d''_{23}, 0, 0) \Rightarrow$ no cycles!

Uni-directional:

In General

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!



Bi-directional:
resolves 2-cycles

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:
resolves 3-cycles
Residual DoF tuple (e.g.) $\mathbf{d}' = (d''_{12}, 0, 0, d''_{23}, 0, 0) \Rightarrow$ no cycles!

Uni-directional:
$\mathbf{d}$ achieved!

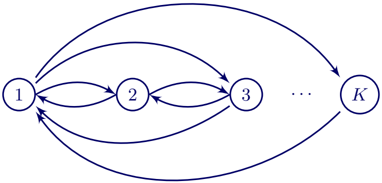
K-user Case

For the K-user Y-channel with $N \leq M$:

K-user Case

For the K-user Y-channel with $N \leq M$:

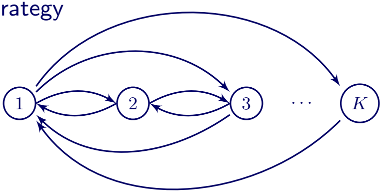
- 2-cycles up to K-cycles,



K-user Case

For the K-user Y-channel with $N \leq M$:

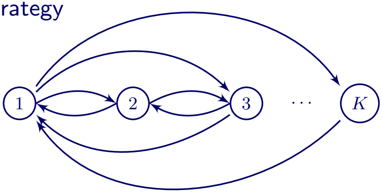
- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy



K-user Case

For the K-user Y-channel with $N \leq M$:

- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy
- exchanges ℓ symbols

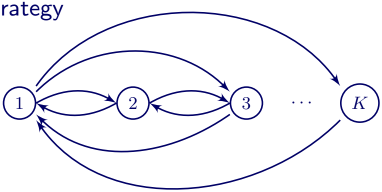


K-user Case

For the K-user Y-channel with $N \leq M$:

- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy

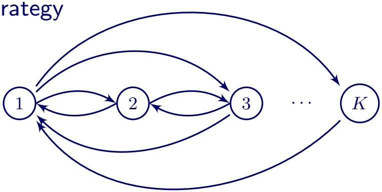
- exchanges ℓ symbols
- requires $\ell - 1$ dimensions



K-user Case

For the K-user Y-channel with $N \leq M$:

- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy

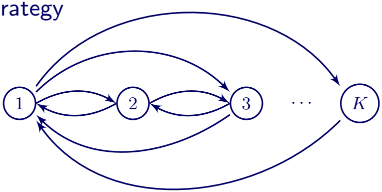


- exchanges ℓ symbols
- requires $\ell - 1$ dimensions
- efficiency $\ell/(\ell - 1)$

K-user Case

For the K-user Y-channel with $N \leq M$:

- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy



- exchanges ℓ symbols
- requires $\ell - 1$ dimensions
- efficiency $\ell/(\ell-1)$

- DoF region described by

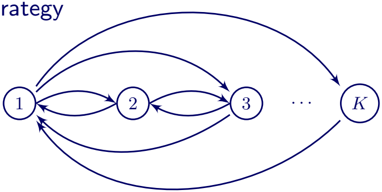
$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d_{p_i p_j} \leq N, \quad \forall \mathbf{p}$$

where $\mathbf{p}$ is a permutation of $(1, 2, \cdots, K)$.

K-user Case

For the K-user Y-channel with $N \leq M$:

- 2-cycles up to K-cycles,
- ℓ-cycles resolved by an ℓ-cyclic strategy
  - exchanges ℓ symbols
  - requires $\ell - 1$ dimensions
  - efficiency $\ell/(\ell - 1)$



- DoF region described by

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d_{p_i p_j} \leq N, \quad \forall \mathbf{p}$$

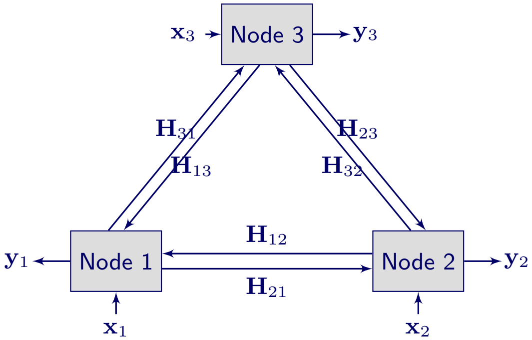
where $\mathbf{p}$ is a permutation of $(1, 2, \cdots, K)$.

Cyclic communication requires joint encoding over multiple sub-channels⇒ MIMO Y-channels are in general inseparable!

Outline

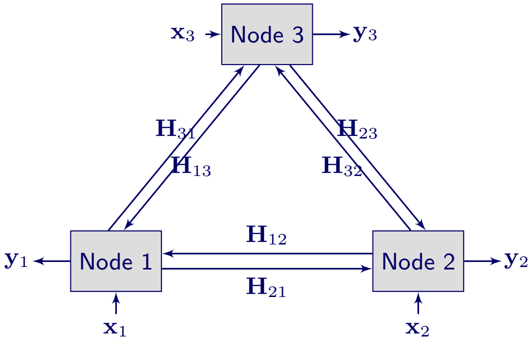
1. From Capacity to DoF
2. MIMO Two-Way Relay Channel
   - Channel diagonalization
   - Signal Alignment
3. MIMO multi-way relay channel
   - Sum-DoF
   - DoF Region
4. MIMO Multi-way Channel

MIMO 3-Way Channel



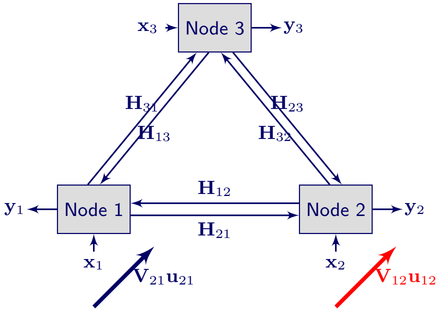
- $\mathbf{y}_k = \mathbf{H}_{kj}\mathbf{x}_j + \mathbf{H}_{ki}\mathbf{x}_i + \mathbf{z}_k$,

MIMO 3-Way Channel



- $\mathbf{y}_k = \mathbf{H}_{kj}\mathbf{x}_j + \mathbf{H}_{ki}\mathbf{x}_i + \mathbf{z}_k,$
- Capacity scaling (DoF)?

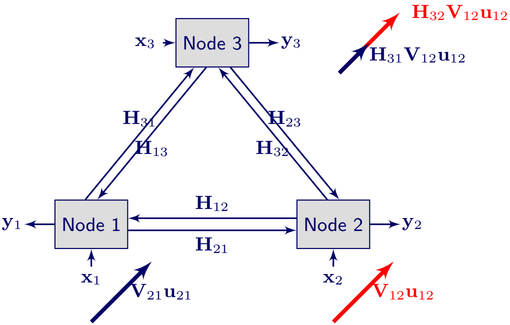
Transmission Scheme



Communication $i \leftrightarrow j$:

- Node i sends $\mathbf{x}_i = \mathbf{V}_{ji}\mathbf{u}_i$, node j sends $\mathbf{x}_j = \mathbf{V}_{ij}\mathbf{u}_j$

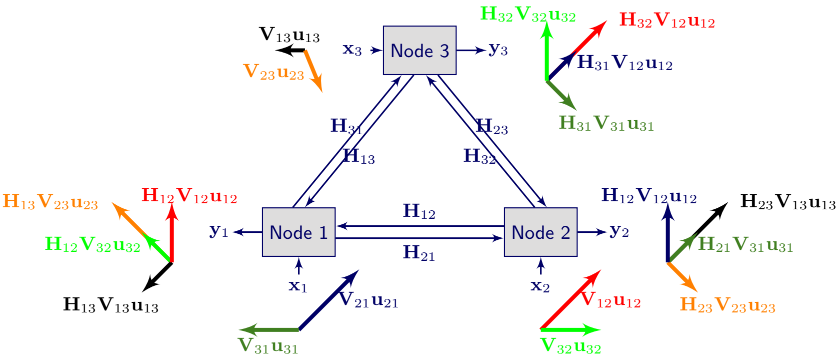
Transmission Scheme



Communication $i \leftrightarrow j$:

- Node i sends $\mathbf{x}_i = \mathbf{V}_{ji}\mathbf{u}_i$, node j sends $\mathbf{x}_j = \mathbf{V}_{ij}\mathbf{u}_j$
- Alignment at node k: Set span$(\mathbf{H}_{ki}\mathbf{V}_{ji}) =$ span$(\mathbf{H}_{kj}\mathbf{V}_{ij})$

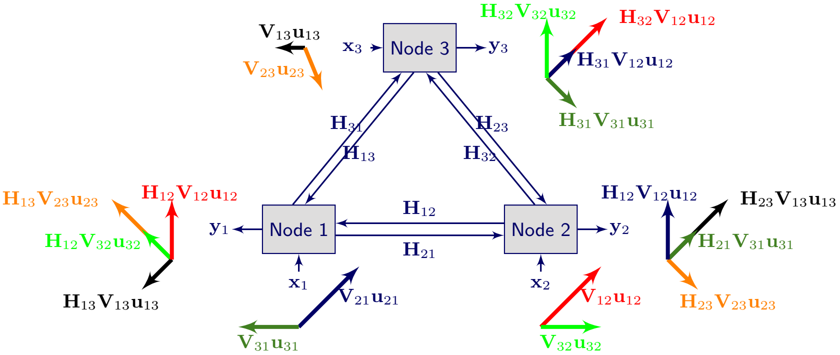
Transmission Scheme



Communication $i \leftrightarrow j$:

- Node i sends $\mathbf{x}_i = \mathbf{V}_{ji}\mathbf{u}_i$, node j sends $\mathbf{x}_j = \mathbf{V}_{ij}\mathbf{u}_j$
- Alignment at node k: Set $\text{span}(\mathbf{H}_{ki}\mathbf{V}_{ji}) = \text{span}(\mathbf{H}_{kj}\mathbf{V}_{ij})$

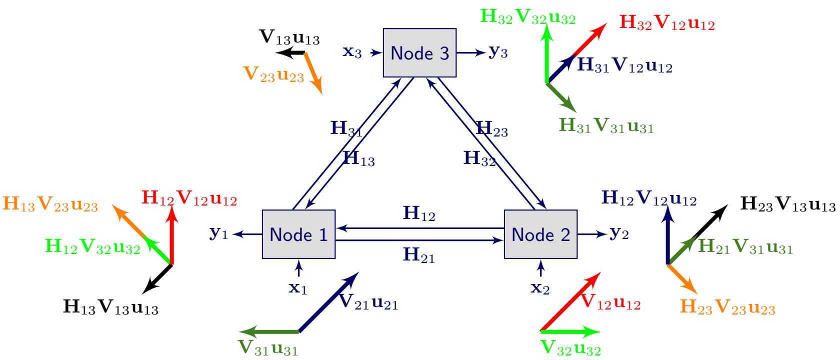
Transmission Scheme



Achievable DoF:

- If $M_1 \geq M_2 \geq M_3$, DoF is $2M_2$

Transmission Scheme



Achievable DoF:

- If $M_1 \geq M_2 \geq M_3$, DoF is $2M_2$
- Optimal scheme (genie-aided bound)

Summary

Two-way Relay Channel:

- Quantize-forward achieves similar rate as CF [Avestimehr *et al.* 10]
- Capacity region known within a constant gap [Nam *et al.* 10]
- Capacity of the BC phase is known [Oechtering *et al.* 08]
- Fading and scheduling [Shaqfeh *et al.* 13]
- Impact of CSIT [Yang *et al.* 13]
- Impact of direct channels [Avestimehr *et al.* 10]
- Energy harvesting [Tutuncuoglu *et al.* 13]
- Multiple relays [Vaze & Heath 09]

Multi-way Relay Channel:

- 3-user LD case with relay private messages, and 4-user LD case [Zewail *et al.* 13]
- Direct links between users [Lee & Heath 13]
- Multi-cast setting: compress-forward [Gündüz *et al.* 13], compute-forward [Ong *et al.* 12]
- Fading case [Wang *et al.* 12]

Multi-way Channel:

- Capacity of classes of 3-way channels [Ong 12]
- Two-way interference channel [Rost 11]
- Two-way IC (feedback better than info. transmission!) [Suh *et al.* 13]
- Two-way networks (MAC,BC,TWC) [Cheng & Devroye 14]

MIMO Two-way Relay Channel:

- Diversity-multiplexing trade-off [Gündüz *et al.* 08], [Vaze & Heath 11],
- Cognition, multiple relays [Alsharoa *et al.* 13],
- Multi-pair sum-rate optimization in [S. *et al.* 09],
- DoF of the K-pair case [Lee & Heath 13], [Cheng & Devroye 13],
- Imperfect CSI [Ubaidulla *et al.* 13], [Zhang *et al.* 13],

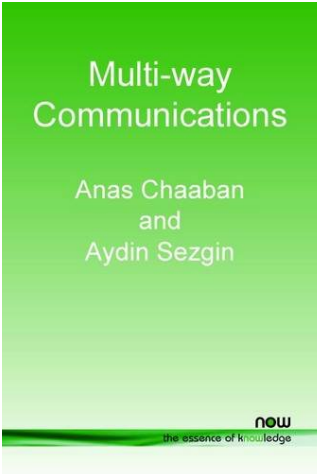
MIMO Multi-way Relay Channel:

- Multi-cluster multi-way relay channels [Tian & Yener 12],
- Performance optimization [Teav *et al.* 14],
- K-user achievable sum-DoF [Lee *et al.* 12],
- Full- vs. half-duplex, global vs. local CSI [Lee & Chun 11],

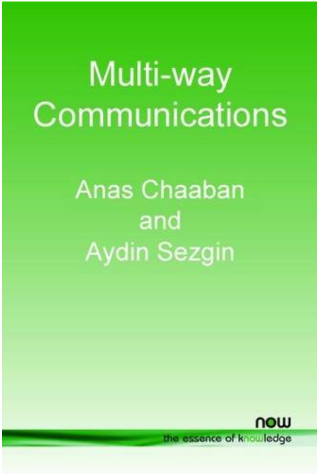
Interesting Problems

- Optimizing the uplink in the two-way relay channel
- Exploiting two-way relaying in larger networks
- Constellations for PLNC and their performance
- Multi-relay cases
- General (approximate) capacity expression for the K-user multi-way relay channel
- Extensions of the multi-way channel to K-users
- Fading multi-way channels
- Capacity region study of the MIMO two-way relay channel
- Sum-DoF of K-user multi-way relay channels (4-user case characterized recently [Wang 14])
- Rate maximization/power minimization,
- Self-interference cancellation techniques and their impact

All this and more to appear in:



All this and more to appear in:



Thank you for your attention!

Optimality

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^{b}}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^{c}}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3} \sum_{j=1,\ j\neq i}^{3} d_{ij}^{u}}^{\text{uni-directional}}$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}}$$

$(d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}}$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c$$

$$(d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31}\ \text{e.g.}\Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^{b}}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^{c}}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^{u}}^{\text{uni-directional}} \qquad (d_{ij}^{u} = d_{ij} - d_{ij}^{b} - d_{ij}^{c})$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^{b} - \sum_{j=2}^{3} d_{1j}^{c} \qquad (d_{ij} + d_{ji} - d_{ij}^{b} = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31}\ \text{e.g.} \Rightarrow d_{13}^{c}=0,\ d_{12}^{b}=d_{21}} - d_{12}^{c} - d_{13}^{c}$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^{c}$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31} \text{ e.g.} \Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^c \qquad (d_{12}^c = d_{12} - d_{12}^b \text{ e.g.})$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31} \text{ e.g.} \Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^c \qquad (d_{12}^c = d_{12} - d_{12}^b \text{ e.g.})$$

$$= d_{12}^b + d_{23} + d_{31}$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31} \text{ e.g.} \Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^c \qquad (d_{12}^c = d_{12} - d_{12}^b \text{ e.g.})$$

$$= d_{12}^b + d_{23} + d_{31}$$

$$= d_{21} + d_{23} + d_{31}$$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31} \text{ e.g.} \Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^c \qquad (d_{12}^c = d_{12} - d_{12}^b \text{ e.g.})$$

$$= d_{12}^b + d_{23} + d_{31}$$

$$= d_{21} + d_{23} + d_{31}$$

No cycles $\Rightarrow N_s \leq N$

Optimality

Total number of dimensions required to achieve $\mathbf{d} \in \mathcal{D}$:

Bi-directional	2 symbols	1 sub-channel
Cyclic	3 symbols	2 sub-channels
Uni-directional	1 symbol	1 sub-channel

$$N_s = \overbrace{\sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b}^{\text{bi-directional}} + \overbrace{\sum_{j=2}^{3} 2d_{1j}^c}^{\text{cyclic}} + \overbrace{\sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij}^u}^{\text{uni-directional}} \qquad (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c)$$

$$= \sum_{i=1}^{3}\sum_{j=1,\ j\neq i}^{3} d_{ij} - \sum_{i=1}^{2}\sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c \qquad (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\})$$

$$= \underbrace{\max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\}}_{d_{12}+d_{23}+d_{31}\ \text{e.g.}\Rightarrow d_{13}^c=0,\ d_{12}^b=d_{21}} - d_{12}^c - d_{13}^c$$

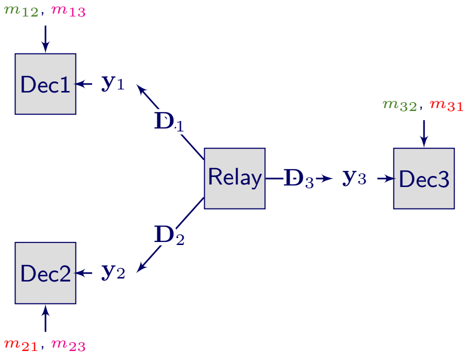
$$= d_{12} + d_{23} + d_{31} - d_{12}^c \qquad (d_{12}^c = d_{12} - d_{12}^b\ \text{e.g.})$$

$$= d_{12}^b + d_{23} + d_{31}$$

$$= d_{21} + d_{23} + d_{31}$$

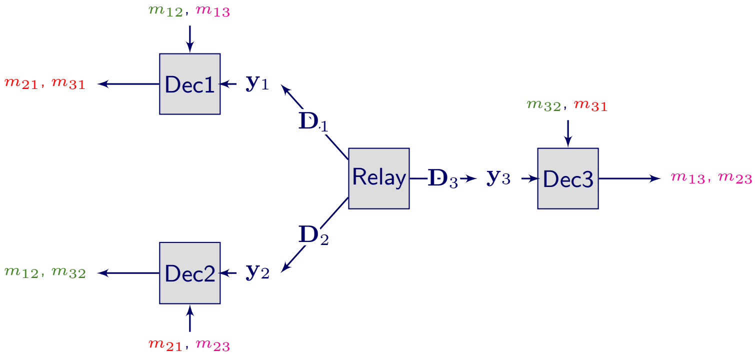
No cycles $\Rightarrow N_s \leq N \Rightarrow$ All $\mathbf{d} \in \mathcal{D}$ are achievable

Outer bound



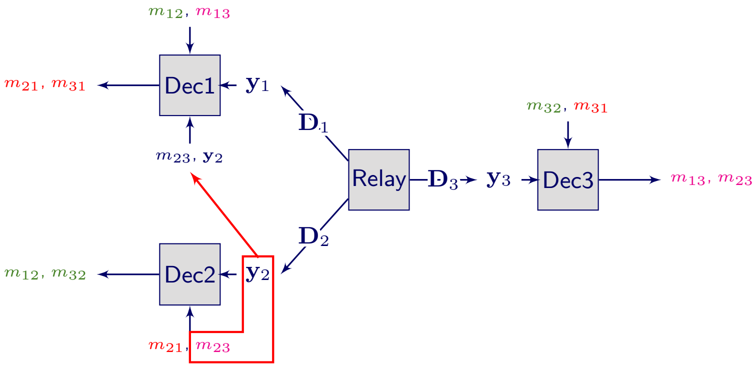
Consider any reliable scheme for the 4-user MIMO MRC

Outer bound



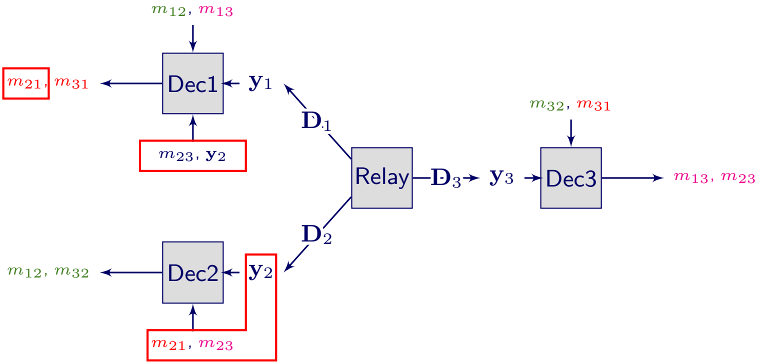
Users can decode their desired signals

Outer bound



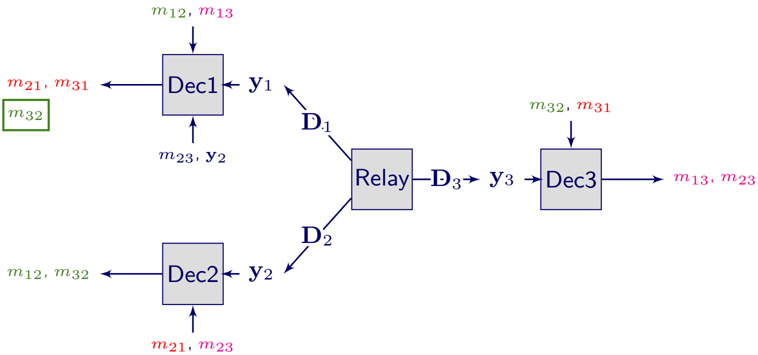
Give m_{23} and $\mathbf{y}_2$ to user 1 as side info.

Outer bound



Now, user 1 has the info. available at user 2

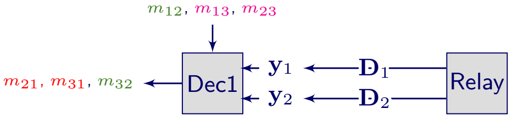
Outer bound



$\Rightarrow$ User 1 can decode m_{32}

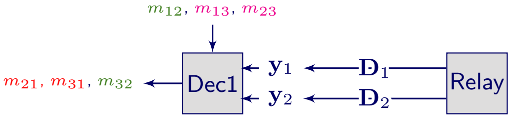
Upper bound

User 1 can decode (m_{21}, m_{31}, m_{32}) from $(m_{12}, m_{13}, \mathbf{y}_1, \overbrace{m_{23}, \mathbf{y}_2}^{\text{side info.}})$



Upper bound

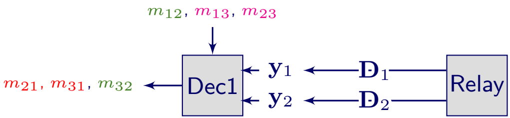
User 1 can decode (m_{21}, m_{31}, m_{32}) from $(m_{12}, m_{13}, \mathbf{y}_1, \overbrace{m_{23}, \mathbf{y}_2}^{\text{side info.}})$



$$\Rightarrow R_{21} + R_{31} + R_{32} \leq I(\mathbf{x}_r; \mathbf{y}_1, \mathbf{y}_2) = I\left(\mathbf{x}_r; \begin{bmatrix}\mathbf{D}_1\\ \mathbf{D}_2\end{bmatrix}\mathbf{x}_r + \begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix}\right)$$ P2P Channel

Upper bound

User 1 can decode (m_{21}, m_{31}, m_{32}) from $(m_{12}, m_{13}, \mathbf{y}_1, \overbrace{m_{23}, \mathbf{y}_2}^{\text{side info.}})$



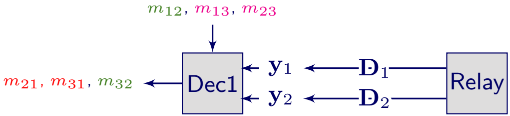
$$\Rightarrow R_{21} + R_{31} + R_{32} \leq I(\mathbf{x}_r; \mathbf{y}_1, \mathbf{y}_2) = I\left(\mathbf{x}_r; \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \mathbf{x}_r + \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}\right)$$

P2P Channel

$$\Rightarrow d_{21} + d_{31} + d_{32} \leq \text{rank}\left(\begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix}\right) = N$$

Upper bound

User 1 can decode (m_{21}, m_{31}, m_{32}) from $(m_{12}, m_{13}, \mathbf{y}_1, \overbrace{m_{23}, \mathbf{y}_2}^{\text{side info.}})$



$$\Rightarrow R_{21} + R_{31} + R_{32} \leq I(\mathbf{x}_r; \mathbf{y}_1, \mathbf{y}_2) = I\left(\mathbf{x}_r; \begin{bmatrix}\mathbf{D}_1 \\ \mathbf{D}_2\end{bmatrix} \mathbf{x}_r + \begin{bmatrix}\mathbf{z}_1 \\ \mathbf{z}_2\end{bmatrix}\right)$$ P2P Channel

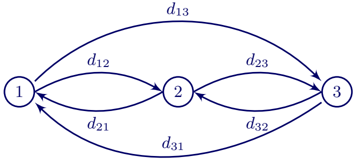
$$\Rightarrow d_{21} + d_{31} + d_{32} \leq \text{rank}\left(\begin{bmatrix}\mathbf{D}_1 \\ \mathbf{D}_2\end{bmatrix}\right) = N$$

Considering different combinations of users gives the desired outer bound

$$\sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{p_i p_j} \leq N, \quad \forall \mathbf{p}$$

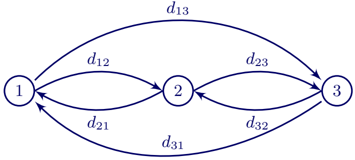
Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$



Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!

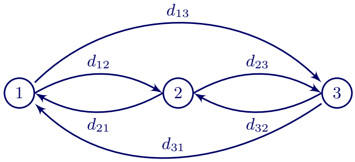


Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$

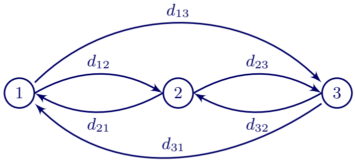


Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels

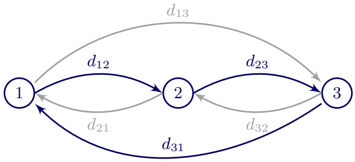


Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:

1) set $d_{ij}^b = d_{ji}^b = \min\{d_{ij}, d_{ji}\}$
2) requires d_{ij}^b sub-channels
3) resolves 2-cycles

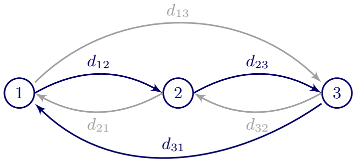


Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$

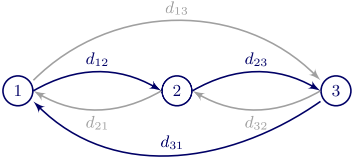


Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ ⇒ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$



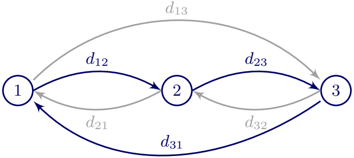
Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ ⇒ 3-cycle!

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$



Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:

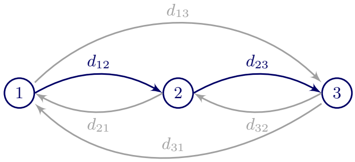
1) set $d^c_{ij} = d^c_{jk} = d^c_{ki} = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ ⇒ 2-cycles and 3-cycles!

Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$



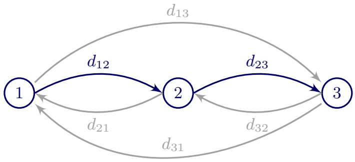
Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ ⇒ 3-cycle!

Cyclic:

1) set $d^c_{ij} = d^c_{jk} = d^c_{ki} = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d^c_{ij}$ sub-channels

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!



Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$

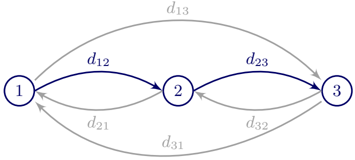
Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:

1) set $d^c_{ij} = d^c_{jk} = d^c_{ki} = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d^c_{ij}$ sub-channels
3) resolves 3-cycles

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ $\Rightarrow$ 2-cycles and 3-cycles!



Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$

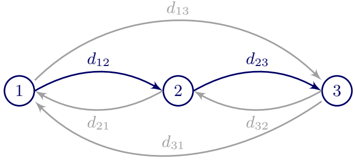
Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ $\Rightarrow$ 3-cycle!

Cyclic:

1) set $d^c_{ij} = d^c_{jk} = d^c_{ki} = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d^c_{ij}$ sub-channels
3) resolves 3-cycles
4) residual DoF $d''_{ij} = d'_{ij} - d^c_{ij}$

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ ⇒ 2-cycles and 3-cycles!



Bi-directional:

1) set $d_{ij}^b = d_{ji}^b = \min\{d_{ij}, d_{ji}\}$
2) requires d_{ij}^b sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d_{ij}^b$

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ ⇒ 3-cycle!

Cyclic:

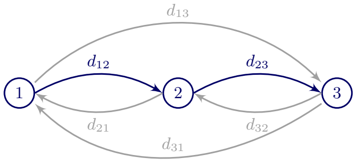
1) set $d_{ij}^c = d_{jk}^c = d_{ki}^c = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d_{ij}^c$ sub-channels
3) resolves 3-cycles
4) residual DoF $d''_{ij} = d'_{ij} - d_{ij}^c$

Uni-directional:

1) set $d_{ij}^u = d''_{ij}$

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32})$ ⇒ 2-cycles and 3-cycles!



Bi-directional:

1) set $d^b_{ij} = d^b_{ji} = \min\{d_{ij}, d_{ji}\}$
2) requires d^b_{ij} sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d^b_{ij}$

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0)$ ⇒ 3-cycle!

Cyclic:

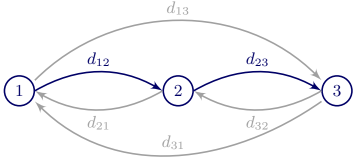
1) set $d^c_{ij} = d^c_{jk} = d^c_{ki} = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d^c_{ij}$ sub-channels
3) resolves 3-cycles
4) residual DoF $d''_{ij} = d'_{ij} - d^c_{ij}$

Uni-directional:

1) set $d^u_{ij} = d''_{ij}$
2) requires d^u_{ij} sub-channels

Resource allocation

Consider a DoF tuple $\mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!



Bi-directional:

1) set $d_{ij}^b = d_{ji}^b = \min\{d_{ij}, d_{ji}\}$
2) requires d_{ij}^b sub-channels
3) resolves 2-cycles
4) residual DoF $d'_{ij} = d_{ij} - d_{ij}^b$

Residual DoF tuple (e.g.) $\mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!

Cyclic:

1) set $d_{ij}^c = d_{jk}^c = d_{ki}^c = \min\{d'_{ij}, d'_{jk}, d'_{ki}\}$
2) requires $2d_{ij}^c$ sub-channels
3) resolves 3-cycles
4) residual DoF $d''_{ij} = d'_{ij} - d_{ij}^c$

Uni-directional:

1) set $d_{ij}^u = d''_{ij}$
2) requires d_{ij}^u sub-channels

$\mathbf{d}$ achieved!