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,

• 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,

- 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,
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
Rapid changes in communications: applications, services, requirements, etc.
Changing Game

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

past          present          future

low-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.

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

Anas Chaaban and Aydin Sezgin
Multi-way Communications
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!

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

**Consequence:** Networks must support much higher data-rates
Ideas

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

⇒ need to study bi-directional communication
Ideas

- Densification of networks
Idea

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

⇒ 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

Important factors: Multi-way communications and Relaying!

⇒ need to study bi-directional communication

Anas Chaaban and Aydin Sezgin
More sophisticated network topologies!

- Machine-to-machine (Point ↔ multi-point)
- Car-to-car (Multi-hop)
- Device-to-device

Important factors: Multi-way communications and Relaying!
More sophisticated network topologies!

- Machine-to-machine

- Point ↔ multi-point

- Car-to-car

- Multi-hop

- Device-to-device

- X-Channel

Important factors: Multi-way communications and Relaying!
More sophisticated network topologies!

- Machine-to-machine
  Point ↔ multi-point

- Car-to-car
  Multi-hop

- Device-to-device

- X-Channel

- Star

Important factors: Multi-way communications and Relaying!
More sophisticated network topologies!

- Machine-to-machine (Point ↔ multi-point)
- Car-to-car (Multi-hop)
- Device-to-device
- X-Channel
- Star
- Interference channel

Important factors: Multi-way communications and Relaying!
More sophisticated network topologies!

- Machine-to-machine: Point ↔ multi-point
- Car-to-car: Multi-hop
- Device-to-device
- X-Channel
- Interference channel

• Important factors: Multi-way communications and Relaying!
More sophisticated network topologies!

- Machine-to-machine (Point↔multi-point)
- Car-to-car (Multi-hop)
- Device-to-device
- X-Channel
- Star
- 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
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],
- Broadcast channel (BC): [Cover 72], [Bergmans 73],
- Interference Channel (IC): [Carleial 75], [Han & Koayashi 81], [Sato 81],
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],
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],
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],
- **Broadcast channel (BC):** [Cover 72], [Bergmans 73],
- **Interference Channel (IC):** [Carleial 75], [Han & Koayashi 81], [Sato 81],
One transmitter, one receiver:
One transmitter, one receiver:

- **Input:** \( x = (x(1), \cdots, x(n)) \) with power \( P \),
One transmitter, one receiver:

- **Input:** $x = (x(1), \cdots, x(n))$ with power $P$,
- **Output:** $y = x + z$, where $z$ is noise,
One transmitter, one receiver:

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

Message

\[ m \in \{1, \cdots, M\} \rightarrow \text{Encoder} \rightarrow x \rightarrow \text{Channel} \rightarrow y \rightarrow \text{Decoder} \rightarrow \hat{m} \]
One transmitter, one receiver:

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

Message

\[ m \in \{1, \cdots, M\} \rightarrow \text{Encoder} \rightarrow x \rightarrow \text{Channel} \rightarrow y \rightarrow \text{Decoder} \rightarrow \hat{m} \]

- **Error probability:** \( P_e = \text{Prob}(\hat{m} \neq m) \),
One transmitter, one receiver:

- **Input:** \( x = (x(1), \ldots, x(n)) \) with power \( P \),
- **Output:** \( y = x + z \), where \( z \) is noise,

Message

\[
m \in \{1, \ldots, M\} \rightarrow \text{Encoder} \rightarrow x \rightarrow \text{Channel} \rightarrow y \rightarrow \text{Decoder} \rightarrow \hat{m}
\]

- **Error probability:** \( P_e = \text{Prob}(\hat{m} \neq m) \),
- **Reliable Comm:** \( P_e \rightarrow 0 \) as \( n \rightarrow \infty \),
One transmitter, one receiver:

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

Message
\[
m \in \{1, \cdots, M\} \rightarrow \text{Encoder} \rightarrow x \rightarrow \text{Channel} \rightarrow y \rightarrow \text{Decoder} \rightarrow \hat{m}
\]

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

![Diagram showing the range of $x$ values from $-\sqrt{P}$ to $\sqrt{P}$]
P2P: Simple example

Let the code-length \( n = 1 \)

Assume:

- \( \text{T}x \) power \( P \), e.g. \( |x| < \sqrt{P} \)
- Noise \( |z| < \sigma \)
- \textbf{Question}: How many points can we place between \(-\sqrt{P}\) and \(\sqrt{P}\) at a distance of \(2\sigma\)?
P2P: Simple example

Let the code-length \( n = 1 \)

Assume:

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

- \( \text{Answer: } M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}} \)
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 \(2\sigma\)?

**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:

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

- \( \text{Answer: } M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}} \) (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:
- \( \text{Tx power } P, \text{ e.g. } |x| < \sqrt{P} \)
- \( \text{Noise } |z| < \sigma \)
- \( \text{Question: How many points can we place between } -\sqrt{P} \text{ and } \sqrt{P} \text{ at a distance of } 2\sigma? \)

- \( \text{Answer: } M = \frac{2\sqrt{P}}{2\sigma} = \sqrt{\text{SNR}} \) (1-D sphere-packing)
- \( M \text{ codewords } \Rightarrow \log_2(M) = \frac{1}{2} \log_2(\text{SNR}) \) bits,
- \( \text{Rate } R = \frac{\log_2(M)}{n} = \frac{1}{2} \log_2(\text{SNR}) \) bits per transmission.
In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).

- With $n = 1$, decoding errors occur,
In reality, Gaussian noise power $\mathbb{E}[z^2] = \sigma^2$ (not $|z| < \sigma$).

- With $n = 1$, decoding errors occur,
- instead, encoding with $n \gg 1$,
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),
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,
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),
  \[ \Rightarrow \text{Rate} = \frac{\log_2(M)}{n} = \frac{1}{2} \log_2(1 + \text{SNR}) \text{ bits per transmission}, \]
- Capacity [Shannon 48].
Two transmitters, one receiver:

- Inputs: $x_1$ and $x_2$
- Powers $P_1$ and $P_2$
- Output: $y = x_1 + x_2 + z$
- Goal: Find the rate-region.

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

Anas Chaaban and Aydin Sezgin
Two transmitters, one receiver:

- Inputs: $x_1$ and $x_2$,
- Powers $P_1$ and $P_2$, 
- Output: $y = x_1 + x_2 + z$,
Two transmitters, one receiver:

- Inputs: $x_1$ and $x_2$,
- Powers $P_1$ and $P_2$,
- Output: $y = x_1 + x_2 + z$. 
Two transmitters, one receiver:

- **Inputs**: $x_1$ and $x_2$,
- **Powers** $P_1$ and $P_2$,
- **Output**: $y = x_1 + x_2 + z$,
- **Goal**: Find the rate-region.
Two transmitters, one receiver:

- **Inputs:** $x_1$ and $x_2$,
- **Powers** $P_1$ and $P_2$,
- **Output:** $y = x_1 + x_2 + z$,
- **Goal:** Find the rate-region.

**Rate-region**

The rate-region is the set of achievable rate pairs $(R_1, R_2)$. 
• Successive decoding:
• Treat $x_2$ as noise $\Rightarrow$

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

  • Treat $x_2$ as noise $\Rightarrow$

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

  • Subtract $x_1$ and decode $x_2$ $\Rightarrow$

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

- Treat \( x_2 \) as noise \( \Rightarrow \)
  \[
  R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{\sigma^2 + P_2} \right),
  \]

- Subtract \( x_1 \) and decode \( x_2 \) \( \Rightarrow \)
  \[
  R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right),
  \]

- opposite order,
• Successive decoding:
• Treat $x_2$ as noise $\Rightarrow$ 
  \[ R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{\sigma^2 + P_2} \right), \]
• Subtract $x_1$ and decode $x_2$ $\Rightarrow$ 
  \[ R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right), \]
• opposite order,
• Capacity region [Ahlswede 71], 
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right), \quad i = 1, 2, \]
  \[ R_1 + R_2 \leq \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right). \]
one transmitter, two receivers:
one transmitter, two receivers:

- **Input:** $x$ with power $P$, 

\[ y_i = x + z_i \]

\[ \sigma_1^2 \leq \sigma_2^2 \]

**Goal:** Find the rate-region, $Rx_1$, $Rx_2$, $Tx$, $y_1$, $y_2$, $z_1$, $z_2$, $x + z$.
one transmitter, two receivers:

- **Input:** \( x \) with power \( P \),
- **Outputs:** \( y_i = x + z_i \),
- **Noises:** \( \sigma_1^2 \leq \sigma_2^2 \),
one transmitter, two receivers:

- **Input**: $x$ with power $P$,
- **Outputs**: $y_i = x + z_i$,
- **Noises**: $\sigma_i^2 \leq \sigma_j^2$,
- **Goal**: Find the rate-region,
BC

- **Superposition coding:**
- **Send** \( x = x_1 + x_2 \), **powers** \( P_1 + P_2 = P \),
Superposition coding:

Send $x = x_1 + x_2$, powers $P_1 + P_2 = P$,

Both receivers decode $x_2$ treating $x_1$ as noise $\Rightarrow$

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

Rate-region = Capacity region \cite{Cover72, Bergmans74}.
• Superposition coding:
• Send $x = x_1 + x_2$, powers $P_1 + P_2 = P$,
• Both receivers decode $x_2$ treating $x_1$ as noise $\Rightarrow$
  $$R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma_2^2 + P_1} \right),$$
• Rx1 subtracts $x_2$ and decodes $x_1$ $\Rightarrow$
  $$R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{\sigma_1^2} \right),$$
• **Superposition coding:**
• Send $x = x_1 + x_2$, powers $P_1 + P_2 = P$,
• Both receivers decode $x_2$ treating $x_1$ as noise $\Rightarrow$
  \[ R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma_2^2 + P_1} \right), \]
• $Rx_1$ subtracts $x_2$ and decodes $x_1$ $\Rightarrow$
  \[ R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{\sigma_1^2} \right), \]
• Rate-region = **Capacity region** [Cover 72], [Bergmans 74],
one transmitter, one receiver, one relay:
 Relay Channel

one transmitter, one receiver, one relay:

- **Inputs:** $x$, $x_r$,
- **Powers:** $P$, $P_r$, 

\[ y = x + x_r + z, \]
\[ y_r = x + z_r. \]
Relay Channel

one transmitter, one receiver, one relay:

- **Inputs:** $x, x_r,$
- **Powers:** $P, P_r,$
- **Outputs:** $y = x + x_r + z,$ $y_r = x + z_r,$
one transmitter, one receiver, one relay:

- **Inputs:** $x$, $x_r$,
- **Powers:** $P$, $P_r$,
- **Outputs:** $y = x + x_r + z$, $y_r = x + z_r$,
- **Noise variance:** $\sigma^2$ and $\sigma_r^2$. 
Relay Channel

- **Main schemes:**
  - Decode-forward,
  - Compress-forward

![Diagram of Relay Channel]

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.
Main schemes: Decode-forward, Compress-forward

Notice: Interaction between consecutive symbols/codewords at Rx!
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 $x \Rightarrow$

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

**Decode-forward:**

- Relay decodes $x$ ⇒
  \[
  R \leq \frac{1}{2} \log \left( 1 + \frac{P}{\sigma_r^2} \right).
  \]

- Relay re-encodes $x$ to $x_r$ and sends it.
**Relay Channel: DF**

![Diagram of Relay Channel]

**Decode-forward:**

- Relay decodes $x$ \(\Rightarrow\)

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

- Relay re-encodes $x$ to $x_r$ 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 $x \Rightarrow R \leq \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2} \right)$.

- Relay re-encodes $x$ to $x_r$ and sends it.

- Rx decodes $x_r \Rightarrow R \leq \frac{1}{2} \log \left( 1 + \frac{P_r}{\sigma^2} \right)$.

- Achievable rate $\min \left\{ \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2} \right), \frac{1}{2} \log \left( 1 + \frac{P_r}{\sigma^2} \right) \right\}$. 
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-set Bound: Capacity bounded by the rate of information flow from a sub-set of nodes to the remaining nodes

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

- **DF Optimal:** Coincides with the cut-set bound.
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 $\Rightarrow C \leq \frac{1}{2} \log \left(1 + \frac{P}{\sigma^2_r}\right)$.
- 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 ⇒ $C \leq \frac{1}{2} \log \left(1 + \frac{P_r}{\sigma^2} \right)$. 
- Cut 2 ⇒ $C \leq \frac{1}{2} \log \left(1 + \frac{P_T}{\sigma^2} \right)$.

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

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

- DF Optimal: Coincides with the cut-set bound.
Relay Channel: CF

Compress-forward:

- Relay compresses/quantizes $y_r$ to $\hat{y}_r = y_r + z_c$ ($z_c$ compression noise)

$\hat{y}_r$-space
Relay Channel: CF

Compress-forward:
- Relay compresses/quantizes $y_r$ to $\hat{y}_r = y_r + z_c$ ($z_c$ compression noise)
Compress-forward:

- Relay compresses/quantizes $y_r$ to $\hat{y}_r = y_r + z_c$ ($z_c$ compression noise)
- Compression rate $R_c \left( \frac{1}{n} \log(\text{number of bins}) \right)$
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 $\hat{y}_r$ to $x_r$ (rate $R_c$).
Compress-forward:

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

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

- Compression noise variance $D = (P + \sigma_r^2) \cdot 2^{-2R_c}$ (optimal rate-distortion)
- Relay encodes $\hat{y}_r$ to $x_r$ (rate $R_c$).
- Rx decodes $x_r \Rightarrow R_c = \frac{1}{2} \log (1 + \frac{P_r}{\sigma_r^2})$.
- Rx obtains $\hat{y}_r = x + z_r + z_c$.
- Rx then decodes $x$ from $\hat{y}_r \Rightarrow R = \frac{1}{2} \log \left(1 + \frac{P}{\sigma_r^2 + D}\right) \leq \frac{1}{2} \log \left(1 + \frac{P(P_r + \sigma_r^2)}{\sigma_r^2(P_r + \sigma_r^2) + \sigma_r^2(P + \sigma_r^2)}\right)$.
Part 2: SISO Bi-directional
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 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: $x_1$ and $x_2$ 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**: $x_1$ and $x_2$ with powers $P_1$ and $P_2$,
- **Outputs**: $y_i = x_j + h_i x_j + z_i$, $h_i \in \mathbb{R}$, $z_i$ has variance $\sigma_i^2$.

![Diagram of two-way channel](image)
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: $x_1$ and $x_2$ with powers $P_1$ and $P_2$,
- Outputs: $y_i = x_j + h_i x_j + z_i$, $h_i \in \mathbb{R}$, $z_i$ has variance $\sigma_i^2$,
- Achievable rate:
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma_j^2} \right) \] (independent of self-interference!),

\begin{align*}
\text{Node 1} & \quad \text{Node 2} \\
\downarrow y_1 & \quad \uparrow x_2 \\

\text{Node 1} & \quad \text{Node 2} \\
\downarrow h_1 & \quad \uparrow h_2 \\
\downarrow z_1 & \quad \uparrow z_2 \\
\end{align*}
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: $x_1$ and $x_2$ with powers $P_1$ and $P_2$,
- Outputs: $y_i = x_j + h_i x_j + z_i$, $h_i \in \mathbb{R}$, $z_i$ has variance $\sigma_i^2$,
- 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.

Node 1

\[ h_1 \]

\[ x_1 \]

\[ y_1 \]

\[ z_1 \]

Node 2

\[ h_2 \]

\[ x_2 \]

\[ y_2 \]

\[ z_2 \]
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**: $x_1$, $x_2$, $x_r$ 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: $x_1, x_2, x_r$ with powers $P_1, P_2,$ and $P_r$,
- Outputs: $y_r = x_1 + x_2 + z_r$, $y_i = x_r + z_i$, noise variance $\sigma^2_j$, 
Two-way Relay Channel

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

Gaussian two-way relay channel:

- Inputs: $x_1, x_2, x_r$ with powers $P_1, P_2$, and $P_r$,
- Outputs: $y_r = x_1 + x_2 + z_r$, $y_i = x_r + z_i$, noise variance $\sigma_j^2$,
- 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 ⇒
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right) \]
  \[ R_1 + R_2 \leq \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right), \]

- Treat downlink as a BC ⇒
  \[ R_1 \leq \frac{1}{2} \log \left( 1 + \frac{p_1}{\sigma^2_1} \right) \]
  \[ R_2 \leq \frac{1}{2} \log \left( 1 + \frac{p_2}{\sigma^2_2 + p_1} \right) \]
  with \( p_1 + p_2 \leq P_r \),

Diagram:
- Node 1
- Relay
- Node 2
- Variables: \( x_1, x_2, y_1, y_2, z_1, z_2, m_1, m_2 \)
- Graph:
  - \( R_1 \) vs. \( R_2 \)
Classical approach

- Treat uplink as a MAC ⇒
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right) \]
  \[ R_1 + R_2 \leq \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right), \]

- Treat downlink as a BC ⇒
  \[ R_1 \leq \frac{1}{2} \log \left( 1 + \frac{p_1}{\sigma_1^2} \right) \]
  \[ R_2 \leq \frac{1}{2} \log \left( 1 + \frac{p_2}{\sigma_2^2 + p_1} \right) \]
  with \( p_1 + p_2 \leq P_r \),

- Achievable region: intersection
Classical approach

- Treat uplink as a MAC ⇒
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right) \]
  \[ R_1 + R_2 \leq \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right), \]

- Treat downlink as a BC ⇒
  \[ R_1 \leq \frac{1}{2} \log \left( 1 + \frac{p_1}{\sigma_1^2} \right) \]
  \[ R_2 \leq \frac{1}{2} \log \left( 1 + \frac{p_2}{\sigma_2^2 + p_1} \right) \]
  with \( p_1 + p_2 \leq P_r \),

- Achievable region: intersection

- Cut-set bound:
  \[ R_i \leq \min \left\{ \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right), \frac{1}{2} \log \left( 1 + \frac{P_r}{\sigma^2} \right) \right\} \]
Classical approach

- Treat uplink as a MAC ⇒
  \[ R_i \leq \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right) \]
  \[ R_1 + R_2 \leq \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right) , \]

- Treat downlink as a BC ⇒
  \[ R_1 \leq \frac{1}{2} \log \left( 1 + \frac{p_1}{\sigma_1^2} \right) \]
  \[ R_2 \leq \frac{1}{2} \log \left( 1 + \frac{p_2}{\sigma_2^2 + p_1} \right) \]
  with \( p_1 + p_2 \leq P_r \),

- Achievable region: intersection

- Cut-set bound:
  \[ R_i \leq \min \left\{ \frac{1}{2} \log \left( 1 + \frac{P_i}{\sigma^2} \right) , \frac{1}{2} \log \left( 1 + \frac{P_r}{\sigma^2} \right) \right\} \]
Sum-rate

Let us focus on the sum-rate $R_{\Sigma} = R_1 + R_2$:
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:** \( R_\Sigma \leq \frac{1}{2} \log(1 + 2P) \),
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: $R_{\Sigma} \leq \frac{1}{2} \log(1 + P),$
Let us focus on the sum-rate $R_\Sigma = R_1 + R_2$:

- **simplifying assumption**: $P_1 = P_2 = P_r = P$, $\sigma^2_1 = \sigma^2_2 = \sigma^2 = 1$,
- **MAC**: $R_\Sigma \leq \frac{1}{2} \log(1 + 2P)$,
- **BC**: $R_\Sigma \leq \frac{1}{2} \log(1 + P)$,

$\Rightarrow \max R_{DF} = \frac{1}{2} \log(1 + P)$
Let us focus on the sum-rate \( R_\Sigma = R_1 + R_2 \):

- **simplifying assumption:**
  \( P_1 = P_2 = P_r = P \),
  \( \sigma^2_1 = \sigma^2_2 = \sigma^2 = 1 \),
- **MAC:** \( R_\Sigma \leq \frac{1}{2} \log(1 + 2P) \),
- **BC:** \( R_\Sigma \leq \frac{1}{2} \log(1 + P) \),

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

- **Question:** How to improve?
Network Coding

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

\[ m_r = m_1 \oplus m_2 \text{ (rate } R_r = \max\{R_1, R_2\} \text{)}, \quad \text{and sends } x_r(m_r), \]

Node $i$ decodes $x_r(m_r)$ and calculates $m_j = m_r \oplus m_i$, ⇒ $R_r \leq \frac{1}{2} \log(1 + 2P) \Rightarrow R_{\Sigma} \leq \log(1 + P)$ ⇒ $R_{NC} = \frac{1}{2} \log(1 + 2P)$ (alternative: BC with side info.).

Good at low SNR, but not at high SNR.

Further improvement?
Network Coding

- Treat uplink as a MAC \( \Rightarrow R_\Sigma \leq \frac{1}{2} \log(1 + 2P) \).
- Relay obtains \( x_1(m_1) \) and \( x_2(m_1) \).
Network Coding

- Treat uplink as a MAC ⇒ $R_{\Sigma} \leq \frac{1}{2} \log(1 + 2P)$.
- Relay obtains $x_1(m_1)$ and $x_2(m_1)$.
- **NC:** Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $x_r(m_r)$,
Network Coding

- Treat uplink as a MAC \( \Rightarrow R_\Sigma \leq \frac{1}{2} \log(1 + 2P) \).
- Relay obtains \( x_1(m_1) \) and \( x_2(m_1) \).
- **NC**: Relay calculates \( m_r = m_1 \oplus m_2 \) (rate \( R_r = \max\{R_1, R_2\} \)), and sends \( x_r(m_r) \),
- Node \( i \) decodes \( x_r(m_r) \) and calculates \( m_j = m_r \oplus m_i \).
Treat uplink as a MAC ⇒ $R_\Sigma \leq \frac{1}{2} \log(1 + 2P)$.

Relay obtains $x_1(m_1)$ and $x_2(m_1)$.

**NC:** Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $x_r(m_r)$,

Node $i$ decodes $x_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,

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

- Treat uplink as a MAC $\Rightarrow R_{\Sigma} \leq \frac{1}{2} \log(1 + 2P)$.
- Relay obtains $x_1(m_1)$ and $x_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $x_r(m_r)$,
- Node $i$ decodes $x_r(m_r)$ 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 ⇒
  \[ R_\Sigma \leq \frac{1}{2} \log(1 + 2P). \]
- Relay obtains \( x_1(m_1) \) and \( x_2(m_1) \).
- **NC:** Relay calculates \( m_r = m_1 \oplus m_2 \) (rate \( R_r = \max\{R_1, R_2\} \)), and sends \( x_r(m_r) \),
- Node \( i \) decodes \( x_r(m_r) \) and calculates \( m_j = m_r \oplus m_i \),
  \[ R_r \leq \frac{1}{2} \log(1 + P) \Rightarrow \]
  \[ R_\Sigma \leq \log(1 + P) \]
  \[ R_{NC} = \frac{1}{2} \log(1 + 2P) \] (alternative: BC with side info.)
- Good at low SNR, but not at high SNR

![Diagram](image-url)
Network Coding

- Treat uplink as a MAC ⇒ $R_\Sigma \leq \frac{1}{2} \log(1 + 2P)$.
- Relay obtains $x_1(m_1)$ and $x_2(m_1)$.
- NC: Relay calculates $m_r = m_1 \oplus m_2$ (rate $R_r = \max\{R_1, R_2\}$), and sends $x_r(m_r)$,
- Node $i$ decodes $x_r(m_r)$ and calculates $m_j = m_r \oplus m_i$,
- ⇒ $R_r \leq \frac{1}{2} \log(1 + P)$ ⇒ $R_\Sigma \leq \log(1 + P)$
- ⇒ $R_{NC} = \frac{1}{2} \log(1 + 2P)$ (alternative: BC with side info.)
- Good at low SNR, but not at high SNR
- Further improvement?

![Network Coding Diagram](attachment:Network_Coding_Diagram.png)
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

![Diagram of node connections](image-url)
**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!
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!
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!

[From Avestimehr et al. 07]

Analyse using the Linear Deterministic (LD) Model
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
The linear-deterministic approximation

\begin{itemize}
  \item Let $x$ and $z$ have unit power and $y = \sqrt{P}x + z,$
\end{itemize}
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

$$y = 2^{\frac{1}{2}\log(P)} \sum_{i=-\infty}^{\infty} x_i2^{-i} + \sum_{i=-\infty}^{\infty} z_i2^{-i}$$

$$\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i2^{-i} + \sum_{i=-\infty}^{\infty} z_i2^{-i} \text{ assuming } |x| \leq 1$$

$$\approx 2^{\frac{1}{2}\log(P)} \sum_{i=1}^{\infty} x_i2^{-i} + \sum_{i=1}^{\infty} z_i2^{-i} \text{ assuming } |z| < 1$$
The linear-deterministic approximation

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

• Write the binary representation

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

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

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

$$\approx 2^n \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \text{ defining } n = \left\lceil \frac{1}{2} \log(P) \right\rceil$$
The linear-deterministic approximation

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

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

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

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

$$\approx 2^n \sum_{i=1}^{\infty} x_i 2^{-i} + \sum_{i=1}^{\infty} z_i 2^{-i} \text{ defining } n = \left\lceil \frac{1}{2} \log(P) \right\rceil$$

$$= 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \text{ regrouping}$$
The linear-deterministic approximation

\[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \]

- noiseless bits
- noisy bits

\[ C \approx \left\lceil \frac{1}{2} \log(P) \right\rceil \]

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 \( x = [x_1, x_2, \cdots, x_q]^T \) and output \( y = S_q - n x \) where \( q \geq n = \left\lceil \frac{1}{2} \log(P) \right\rceil \) and \( S \) is a down-ward shift matrix.
The linear-deterministic approximation

\[
y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i},
\]

- noiseless bits
- noisy bits

• Ignore the noisy bits \(\Rightarrow\)

\[
y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i}
\]
The linear-deterministic approximation

\[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \]

- Ignore the noisy bits
  \[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} \]
  \[ \Rightarrow \text{Channel can be modelled as } n \text{ bit-pipes!} \]
The linear-deterministic approximation

\[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \]

- 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!
  \( \Rightarrow \) \( C \approx n = \lceil \frac{1}{2} \log(P) \rceil \)

Impact of noise modelled by clipping the least-significant bits
The linear-deterministic approximation

\[
y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i},
\]

- 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!
  \( \Rightarrow \) \( C \approx n = \lceil \frac{1}{2} \log(P) \rceil \)

- A good approximation at high SNR

Impact of noise modelled by clipping the least-significant bits
The linear-deterministic approximation

\[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} + \sum_{i=1}^{\infty} (x_{i+n} + z_i) 2^{-i}, \]

- Ignore the noisy bits ⇒ 
  \[ y \approx 2^n \sum_{i=1}^{n} x_i 2^{-i} \]
  
  ⇒ Channel can be modelled as \( n \) bit-pipes!
  
  ⇒ \( C \approx n = \lceil \frac{1}{2} \log(P) \rceil \)

- A good approximation at high SNR

**Deterministic P2P**

A Gaussian P2P can be approximated as a binary channel with input \( x = [x_1, x_2, \cdots, x_q]^T \) and output \( y = S^{q-n} x \) where \( q \geq n = \lceil \frac{1}{2} \log(P) \rceil \) and

\[ S = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \text{ is a down-ward shift matrix.} \]
The linear-deterministic approximation

Similar approximation can be applied to the:

- **MAC:**
  \[ y = S^{q - n_1} x_1 \oplus S^{q - n_2} x_2 \]
  where \( n_i = \left\lceil \frac{1}{2} \log(P_i) \right\rceil \) and \( q = \max\{n_1, n_2\} \),

- **BC:**
  \[ y_i = S^{q - n_i} x_i \]
The linear-deterministic approximation

Similar approximation can be applied to the:

- **MAC:**
  \[ y = S^{q-n_1} x_1 \oplus S^{q-n_2} x_2 \]
  where \( n_i = \lceil \frac{1}{2} \log(P_i) \rceil \) and \( q = \max\{n_1, n_2\} \),

- **BC:**
  \[ y_i = S^{q-n_i} x_i \]
The linear-deterministic approximation

Similar approximation can be applied to the:

- **MAC:**
  \[ y = S^{q-n_1} x_1 \oplus S^{q-n_2} x_2 \]
  where \( n_i = \left\lceil \frac{1}{2} \log(P_i) \right\rceil \) and \( q = \max\{n_1, n_2\} \),

- **BC:**
  \[ y_i = S^{q-n_i} x \]
Similar approximation can be applied to the:

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

- **BC:** \( y_i = \mathbf{S}^{q-n_i} \mathbf{x} \),
The linear-deterministic approximation

Similar approximation can be applied to the:

1. **MAC:**
   \[ y = S^{q-n_1} x_1 \oplus S^{q-n_2} x_2 \]
   where \( n_i = \left\lceil \frac{1}{2} \log(P_i) \right\rceil \) and 
   \( q = \max\{n_1, n_2\} \),

2. **BC:** \( y_i = S^{q-n_i} x \),

3. A very useful tool for studying Gaussian networks

MAC with \( n_1 \geq n_2 \)

MAC: transmit 1 to 2, 2 to 1

BC with \( n_1 \geq n_2 \)

BC: transmit 1 to 2, 2 to 1

Lost bit at Rx2

Anas Chaaban and Aydin Sezgin
Bi-Directional Communications
Similar approximation can be applied to the:

- **MAC:**
  
  \[ y = S^{q-n_1} x_1 \oplus S^{q-n_2} x_2 \]

  where \( n_i = \lceil \frac{1}{2} \log(P_i) \rceil \) and \( q = \max\{n_1, n_2\} \),

- **BC:** \( y_i = S^{q-n_i} x \),

- A very useful tool for studying Gaussian networks

- Obtained insights in an LD network can be extended to corresponding Gaussian networks
LD Two-way relay channel

- **Uplink**: MAC with 
  \[ n_1 = n_2 = n = 4, \]
- **Node** \( i \) sends \( x_i \in \mathbb{F}_2^q \),
- **Relay** receives \( S^{q-n} x_1 \oplus S^{q-n} x_2 \),
- \( x_1 \oplus x_2 \) 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 $x_i \in \mathbb{F}_2^q$,
- **Relay** receives $S^{q-n}x_1 \oplus S^{q-n}x_2$,
- $x_1 \oplus x_2$ 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 $x_r = x_1 \oplus x_2$ with rate $R_r = \max\{R_1, R_2\}$,
- **Node** $i$ receives $S^{q-n}x_r$,
- $x_r$ 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** \( x_i \in \mathbb{F}_2^q \),
- **Relay receives** \( S^{q-n} x_1 \oplus S^{q-n} x_2 \),
- **\( x_1 \oplus x_2 \) 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** \( x_r = x_1 \oplus x_2 \) with rate \( R_r = \max\{R_1, R_2\} \),
- **Node \( i \) receives** \( S^{q-n} x_r \),
- **\( x_r \) 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 $\max\{R_1, R_2\} \leq n = \lceil \frac{1}{2} \log(P) \rceil$. 
**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 $\max\{R_1, R_2\} \leq n = \lceil \frac{1}{2} \log(P) \rceil$.
- $R_{CF} = 2n = 2 \lceil \frac{1}{2} \log(P) \rceil \approx \log(P)$
- vs. $R_{NC} = \frac{1}{2} \log(1 + 2P)$ for network coding!
LD Two-way relay channel

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

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?
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

- $\mathbf{x}$: Fine lattice $\Lambda_f$
Nested-lattice codes

- $\mathbf{x}$: Fine lattice $\Lambda_f$
- Coarse lattice $\Lambda_c \subset \Lambda_f$
Nested-lattice codes

- $x$: Fine lattice $\Lambda_f$
- Coarse lattice $\Lambda_c \subset \Lambda_f$
- Modulo $\Lambda_c$
Nested-lattice codes

- $x$: Fine lattice $\Lambda_f$
- Coarse lattice $\Lambda_c \subset \Lambda_f$
- Modulo $\Lambda_c$
- Nested lattice code
Nested-lattice codes

- \( \mathbf{x} \): Fine lattice \( \Lambda_f \)
- Coarse lattice \( \Lambda_c \subset \Lambda_f \)
- Modulo \( \Lambda_c \)
- Nested lattice code
- Power Constraint satisfied by the choice of \( \Lambda_c \)
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 $x_1$ and $x_2$,
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power $P$,
- send $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$, 

Remarks:
- $x_r$ belongs to the same nested lattice codebook $\Rightarrow R_r = R$ (same rate as $x_1$ and $x_2$)
- Computation rate $[\text{Nazer & Gastpar 11}]$
  Relay can compute $(x_1 + x_2) \mod \Lambda_c$ as long as $R \leq \left[ \frac{1}{2} \log \left( 1 + \frac{P}{2} \right) \right]$. 

Anas Chaaban and Aydin Sezgin
Bi-Directional Communications 24
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power $P$,
- send $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$,
- calculates $y'_r = y_r \mod \Lambda_c$. 

Remarks:
- $x_r$ belongs to the same nested lattice codebook $\Rightarrow R_r = R$ (same rate as $x_1$ and $x_2$)
- Computation rate \cite{Nazer & Gastpar 11}
- Relay can compute $(x_1 + x_2) \mod \Lambda_c$ as long as $R \leq \left[\frac{1}{2}\log(1+P)\right] + \frac{1}{2}$. 

Anas Chaaban and Aydin Sezgin
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power $P$,
- send $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$,
- calculates $y'_r = y_r \mod \Lambda_c$,
- 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 $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$,
- calculates $y'_r = y_r \mod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $x_r = (x_1 + x_2) \mod \Lambda_c$
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power $P$,
- send $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$,
- calculates $y'_r = y_r \mod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $x_r = (x_1 + x_2) \mod \Lambda_c$

Remarks:
- $x_r$ belongs to the same nested lattice codebook
  $\Rightarrow R_r = R$ (same rate as $x_1$ and $x_2$)
Back to the two-way relay channel

- Nodes use nested lattice codes with rate $R_1 = R_2 = R$ and power $P$,
- send $x_1$ and $x_2$,
- Relay receives $y_r = x_1 + x_2 + z_r$,
- calculates $y'_r = y_r \mod \Lambda_c$,
- decode to the nearest codeword,
- relay obtains $x_r = (x_1 + x_2) \mod \Lambda_c$

Remarks:
- $x_r$ belongs to the same nested lattice codebook
  $\Rightarrow R_r = R$ (same rate as $x_1$ and $x_2$)

Computation rate [Nazer & Gastpar 11]

Relay can compute $(x_1 + x_2) \mod \Lambda_c$ as long as $R \leq \left[ \frac{1}{2} \log \left( \frac{1}{2} + P \right) \right]^+$. 

Remarks:
- $x_r$ belongs to the same nested lattice codebook
Back to the two-way relay channel

- Relay sends $x_r$ (rate $R$),

\[
R \leq \frac{1}{2} \log(1 + \frac{P}{2})
\]
Back to the two-way relay channel

- Relay sends $x_r$ (rate $R$),
- node $i$ decodes $x_r$ $\Rightarrow$
  \[ R \leq \frac{1}{2} \log(1 + P). \]
Back to the two-way relay channel

- Relay sends $x_r$ (rate $R$),
- node $i$ decodes $x_r$ ⇒ $R \leq \frac{1}{2} \log(1 + P)$.
- Can $x_j$ be recovered from $x_r$?

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

$$R_{CF} = \left[ \log \left( \frac{1}{2} + P \right) \right] + \text{downlink}$$
Back to the two-way relay channel

- Relay sends $x_r$ (rate $R$),
- Node $i$ decodes $x_r$ \(\Rightarrow R \leq \frac{1}{2} \log(1 + P)\).
- Can $x_j$ be recovered from $x_r$?
- Yes! Node $i$ calculates \((x_r - x_i) \mod \Lambda_c\),

\[
R_{\text{CF}} = \left\lfloor \log\left(1 + P\right) \right\rfloor + R_{\text{uplink}} \\
R_{\text{CF}} = \left\lfloor \frac{1}{2} \log(1 + P) \right\rfloor + R_{\text{downlink}}
\]
Back to the two-way relay channel

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

\[
R \leq \left[\frac{1}{2} \log \left(\frac{1}{2} + P\right)\right]^+ \quad \text{uplink}
\]
\[
R \leq \frac{1}{2} \log (1 + P) \quad \text{downlink}
\]
Back to the two-way relay channel

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

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

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

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

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

$$\begin{align*}
R_1, R_2 &\leq \frac{1}{2} \log \left( 1 + \frac{P_2}{P_1} \right) \\
R_1, R_2 &\leq \frac{1}{2} \log \left( 1 + \frac{P_r}{N} \right)
\end{align*}$$

Time-sharing NC and CF,

Far from sum-capacity!

Can we do better?
CF rate region

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

• Reduce $P_1$ to $P_2$ and use CF:

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

$$R_1, R_2 \leq \frac{1}{2} \log (1 + P_r)$$
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**,
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?
• $P_1 \geq P_2 \Rightarrow n_1 \geq n_2$,
• 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 $R_2 \leq n_2$
- $P_1 \geq P_2 \Rightarrow n_1 \geq n_2$,
- 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 $x_1$,
- node 1 gets $x_2$,
- $R_1$ 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 $R_2 \leq \min\{n_2, n_r\}$.

  $\Rightarrow R_1 \leq \left\lceil \frac{1}{2} \log(\min\{P_1, P_r\}) \right\rceil$, $R_2 \leq \left\lceil \frac{1}{2} \log(\min\{P_2, P_r\}) \right\rceil$,
Achievable rates: \( R_1 \leq \min\{n_1, n_r\} \) and \( R_2 \leq \min\{n_2, n_r\} \).

\[ R_1 \leq \left\lceil \frac{1}{2} \log\left(\min\{P_1, P_r\}\right) \right\rceil, \quad R_2 \leq \left\lceil \frac{1}{2} \log\left(\min\{P_2, P_r\}\right) \right\rceil, \]

- 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 \( R_2 \leq \min\{n_2, n_r\} \).

\[
R_1 \leq \left\lceil \frac{1}{2} \log(\min\{P_1, P_r\}) \right\rceil, \quad R_2 \leq \left\lceil \frac{1}{2} \log(\min\{P_2, P_r\}) \right\rceil,
\]

- asymmetric rates can be achieved by combining CF and DF

How to extend to Gaussian two-way relay channels?
Combination of CF and DF

- **Node 1:** \( x_1 = \sqrt{P_d} x_{1d} + \sqrt{P_c} x_{1c} \)
Combination of CF and DF

- Node 1: \( x_1 = \sqrt{P_d}x_{1d} + \sqrt{P_c}x_{1c} \)
- Node 2: \( x_2 = \sqrt{P_c}x_{2c} \)
Combination of CF and DF

- Node 1: \( x_1 = \sqrt{P_d}x_{1d} + \sqrt{P_c}x_{1c} \)
- Node 2: \( x_2 = \sqrt{P_c}x_{2c} \)
- Powers: \( P_c + P_d \leq P_1, \ P_c \leq P_2. \)
CF rate region

Combination of CF and DF

- Node 1: $x_1 = \sqrt{P_d}x_{1d} + \sqrt{P_c}x_{1c}$
- Node 2: $x_2 = \sqrt{P_c}x_{2c}$
- Powers: $P_c + P_d \leq P_1$, $P_c \leq P_2$.
- Relay receives:
  
  $y_r = \sqrt{P_d}x_{1d} + \sqrt{P_c}(x_{1c} + x_{2c}) + z_r$,
**CF rate region**

**Combination of CF and DF**

- **Node 1:** $x_1 = \sqrt{P_d}x_{1d} + \sqrt{P_c}x_{1c}$
- **Node 2:** $x_2 = \sqrt{P_c}x_{2c}$
- **Powers:** $P_c + P_d \leq P_1$, $P_c \leq P_2$.
- **Relay receives:**
  
  $$y_r = \sqrt{P_d}x_{1d} + \sqrt{P_c}(x_{1c} + x_{2c}) + z_r,$$

- **decodes** $x_{rd} = x_{1d}$ followed by
  
  $$x_{rc} = x_{1c} + x_{2c},$$

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

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

\[ x_r = \sqrt{P_{rc}} x_{rc} + \sqrt{P_{rd}} x_{rd}, \]

- Both nodes decode \( x_{rc} \) and extract desired CF signal, and node 2 decodes \( x_{rd} \).
\( x_r = \sqrt{P_{rc}} x_{rc} + \sqrt{P_{rd}} x_{rd}, \)

- **Powers:** \( P_{rc} + P_{rd} \leq P_r, \)

\( x_r \)
**CF rate region**

- \( x_r = \sqrt{P_{rc}}x_{rc} + \sqrt{P_{rd}}x_{rd}, \)
- **Powers:** \( P_{rc} + P_{rd} \leq P_r, \)
- **Node** \( i \) **receives**
  \( y_i = \sqrt{P_{rc}}x_{rc} + \sqrt{P_{rd}}x_{rd} + z_i \)
CF rate region

- \( x_r = \sqrt{P_{rc}}x_{rc} + \sqrt{P_{rd}}x_{rd} \),
- Powers: \( P_{rc} + P_{rd} \leq P_r \),
- Node \( i \) receives \( y_i = \sqrt{P_{rc}}x_{rc} + \sqrt{P_{rd}}x_{rd} + z_i \)
- both nodes decode \( x_{rc} \), 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 \( \mathbf{x}_{rd} \)

\[
R_c \leq \frac{1}{2} \log \left( 1 + \frac{P_{rc}}{1 + P_{rd}} \right)
\]
\[
R_d \leq \frac{1}{2} \log (1 + P_{rd})
\]
Combining CF and DF achieves $R_1 = R_c + R_d$, $R_2 = R_c$, where

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

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

for $P_c + P_d \leq P_1$, $P_c \leq P_2$, $P_{rc} + P_{rd} \leq P_r$. 
Combining CF and DF achieves $R_1 = R_c + R_d$, $R_2 = R_c$, where

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

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

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

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

Anas Chaaban and Aydin Sezgin

Bi-Directional Communications
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 \(x_{i_kd} + x_{i_kc}\) and the other \(x_{j_kc}\).
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 \(x_{i_k,d} + x_{i_k,c}\) and the other \(x_{j_k,c}\),
- Relay decodes \(x_{i_k,d}\) then \(x_{i_k,c} + x_{j_k,c}\) 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

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
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

![Graph with upper-bound and CF curves](image)
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 ↔ 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 $R_{13} = R_{21} = 2$, $R_{32} = 1 \Rightarrow$

\[
\begin{align*}
\text{Node 1} & \quad \text{Relay} & \quad \text{Node 2} & \quad \text{Node 3} \\
\text{1} & \quad \text{2} & \quad \text{2} & \quad \text{3} \\
\text{2} & \quad \text{2} & \quad \text{1} & \\
\text{1} & \quad \text{2} & \quad 1 & \quad \text{3} \\
\end{align*}
\]
Multi-way Relay Channel

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

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

- Relay has only 4 remaining!

$\Rightarrow$ Achievability requires more CF
Multi-way Relay Channel

- Bi-directional $2 \leftrightarrow 3$

![Diagram of a bi-directional relay channel with nodes 1, 2, and 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** $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$

- Achieves $r_{13} = r_{32} = r_{21} = 1$

- Remainder $R_{13} = R_{21} = 1$

- **Uni-directional** $1 \rightarrow 3$ and $2 \rightarrow 1$: requires 2 bit-pipes

- Relay has 2 remaining
Multi-way Relay Channel

- **Bi-directional** 2 ↔ 3
- **Cyclic** 1 → 3 → 2 → 1:
- Achieves \( r_{13} = r_{32} = r_{21} = 1 \)
- Remainder \( R_{13} = R_{21} = 1 \)
- **Uni-directional** 1 → 3 and 2 → 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]
Gaussian case:

- node $i$ sends $x_{ib} + x_{ic} + x_{iu}$ (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 $j$. 

\[
W_{31}, W_{32} \quad \rightarrow \quad \text{Node 3} \quad \text{Node 1} \quad \text{Node 2}
\]

\[
W_{12}, W_{13} \quad \leftarrow \quad \text{Node 1} \quad \text{Node 2}
\]

\[
W_{21}, W_{23} \quad \leftarrow \quad \text{Node 2}
\]
3-Way Channel (3WC)

- Full message-exchange: Message $W_{ij}$ from node $i$ to $j$,
- Tx signal: $x_j$, power $P$. 

\[ W_{31}, W_{32} \rightarrow \text{Node 3} \]

\[ W_{12}, W_{13} \]

\[ W_{21}, W_{23} \]

\[ x_1 \]

\[ x_2 \]

\[ x_3 \]
3-Way Channel (3WC)

- Full message-exchange: Message $W_{ij}$ from node $i$ to $j$,
- Tx signal: $x_j$, power $P$,
- Rx signal: $y_k = h_i x_j + h_j x_i + z_k$ (reciprocal channels, unit noise power)
3-Way Channel (3WC)

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

$$W_{31}, W_{32} \rightarrow \text{Node 3} \rightarrow y_3
\text{Node 1} \quad \text{Node 2}
\quad x_1, x_2, x_3
\quad h_1, h_2, h_3
\quad W_{12}, W_{13}
\quad W_{21}, W_{23}$$
3-Way Channel (3WC)

- Full message-exchange: Message $W_{ij}$ from node $i$ to $j$,
- Tx signal: $x_j$, power $P$,
- Rx signal: $y_k = h_i x_j + h_j x_i + z_k$ (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 $\log(P)$

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

$$\log(1 + h_3^2P) \leq C_\Sigma \leq \log(1 + h_3^2P) + 2.$$
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!
How to find the capacity region?

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

Assume $h_2 = 0 \Rightarrow$ Y-channel! 

What if $h_2 > 0$?
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!
  - Capacity achieving scheme for the Y-channel is capacity achieving for the 3-way channel
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 achieving scheme for the Y-channel is capacity achieving for the 3-way channel
- What if $h_1^2 > 0$?
If $h_1^2 > 0$:

- Interference between nodes 2 and 3 (w.r.t. Y-channel scheme)
If $h_2^2 > 0$:

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

If $h_2^2 > 0$:

- 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:
  \[ y_3(B) = h_2x_1(B) + h_1x_{23}(B) + z_3(B), \quad y_3(B+1) = h_2x_1(B+1) + z_3(B+1) \]
- After decoding desired signals from $x_1(B+1)$, node 3 removes interference from $x_{23}(B)$ (Causality)
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**:
  \[ y_3(B) = h_2x_1(B) + h_1x_{23}(B) + z_3(B), \quad y_3(B+1) = h_2x_1(B+1) + z_3(B+1) \]

- After decoding desired signals from $x_1(B + 1)$, node 3 removes interference from $x_{23}(B)$ (**Causality**)
- A desired signal at 1: **Interference neutralization**
  \[ y_3 = h_2x_1 + h_1x_{21} + z_3 \]

- Node 2 pre-transmits a signal for interference neutralization:
  \[ x_1 = x_1' - \frac{h_1}{h_2}x_{21} \]
Capacity Region

Main ingredients

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

![Diagram showing communication between nodes with channel coefficients](https://via.placeholder.com/150)
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],
D2D communications: Impact of (D2D) channel

Sym. Rate

- no D2D
- traditional (separate UL/DL)
- 3-way
- upper bound

$h_3$

Anas Chaaban and Aydin Sezgin
Part 3: MIMO
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
From Capacity to Capacity Region

**Single-user (MIMO P2P):**

\[
\begin{array}{c}
\text{Input covariance } Q, \quad \text{tr}(Q) \leq P \\
\end{array}
\]
From Capacity to Capacity Region

Single-user (MIMO P2P):

\[ C = \log |I + HQH^H| \]

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

Achievable rate region
From Capacity to Capacity Region

**Single-user (MIMO P2P):**

\[
\begin{align*}
\text{Tx} & \xrightarrow{H} \text{Rx} \\
\text{Input covariance } Q, \quad \text{tr}(Q) & \leq P
\end{align*}
\]

**Capacity:** 
\[ C = \log |I + HQH^H| \]

**Multi-user (MIMO MAC):**

\[
\begin{align*}
\text{Tx 1} & \xrightarrow{H_1} \text{Rx} \\
\text{Tx 2} & \xrightarrow{H_2} \text{Rx}
\end{align*}
\]

\[
\text{Input covariance } Q_i, \quad i = 1, 2, \quad \text{tr}(Q_i) \leq P
\]
From Capacity to Capacity Region

**Single-user (MIMO P2P):**

\[
\begin{align*}
\text{Capacity: } C &= \log |I + HQH^H| \\
\text{Input covariance } Q, \quad \text{tr}(Q) &\leq P
\end{align*}
\]

**Multi-user (MIMO MAC):**

\[
\begin{align*}
\text{Capacity region: } R_i &\leq \log |I + H_i Q_i H_i^H|, \\
\text{Input covariance } Q_i, \quad i = 1, 2, \quad \text{tr}(Q_i) &\leq P
\end{align*}
\]
From Capacity to Capacity Region

**Single-user (MIMO P2P):**

\[
\text{Capacity: } C = \log |I + HQH^H| 
\]

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

**Multi-user (MIMO MAC):**

Capacity region:

\[
R_i \leq \log |I + H_i Q_i H_i^H|, \\
R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| 
\]

Input covariance \(Q_i, i = 1, 2\), \(\text{tr}(Q_i) \leq P\)
From Capacity to Capacity Region

**Single-user (MIMO P2P):**

\[ C = \log |I + HQH^H| \]

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

**Multi-user (MIMO MAC):**

\[ R_i \leq \log |I + H_i Q_i H_i^H|, \]
\[ R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \]

Input covariance \( Q_i, i = 1, 2, \text{tr}(Q_i) \leq P \)
Capacity to DoF

**Single-user (MIMO P2P):**

\begin{align*}
\text{Tx} & \quad H \quad \text{Rx} \\
\end{align*}
Single-user (MIMO P2P):

\[
\begin{align*}
\text{Capacity:} & \quad C = \log |I + H Q H^H| \\
\end{align*}
\]

\[
\begin{align*}
\text{Optimization: water-filling} & \\
\end{align*}
\]

\[
\begin{align*}
\text{DoF:} & \quad \text{• } M \text{ Tx antennas, } N \text{ Rx antennas} \\
& \quad \Rightarrow H \in C_{N \times M} \\
& \quad \Rightarrow \text{rank}(H) = \min\{M, N\} \\
& \quad \Rightarrow C \approx \min\{M, N\} C(P) \text{ at high } P \quad \text{C(P)} \\
& \quad \text{• } \text{DoF: } d = \lim_{P \to \infty} C(P) = \text{rank}(H) \\
& \quad \Rightarrow d = \min\{M, N\} \\
& \quad \Rightarrow \text{Capacity equivalent to that of } d \text{ parallel SISO P2P channels!} \\
\end{align*}
\]
Capacity to DoF

Single-user (MIMO P2P):

\[
\begin{align*}
\text{Capacity:} \\
C &= \log |I + HQH^H| \\
\text{Optimization: water-filling}
\end{align*}
\]
Capacity to DoF

Single-user (MIMO P2P):

\[
\begin{array}{c}
\text{Tx} \\
\text{H} \\
\text{Rx}
\end{array}
\]

Capacity:

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

Optimization: water-filling

DoF:

- \( M \) Tx antennas, \( N \) Rx antennas
Capacity to DoF

Single-user (MIMO P2P):

\[
\begin{array}{c}
\text{Tx} \\
\hspace{1cm} H \\
\text{Rx}
\end{array}
\]

Capacity:

\[
C = \log |I + HQH^H|
\]

Optimization: water-filling

DoF:

- \( M \) Tx antennas, \( N \) Rx antennas

\[
\Rightarrow H \in \mathbb{C}^{N \times M} \Rightarrow \text{rank}(H) = \min\{M, N\}
\]
Capacity to DoF

Single-user (MIMO P2P):

\[
\begin{align*}
\text{Capacity:} & \quad C = \log |I + \mathbf{H} \mathbf{Q} \mathbf{H}^H| \\
\text{Optimization: water-filling}
\end{align*}
\]

DoF:

- \( M \) Tx antennas, \( N \) Rx antennas
  \[ \Rightarrow \mathbf{H} \in \mathbb{C}^{N \times M} \Rightarrow \text{rank}(\mathbf{H}) = \min\{M, N\} \]
  \[ \Rightarrow C \approx \min\{M, N\} C(P) \quad \text{at high } P \]

\( C(P) \): SISO P2P capacity
Capacity to DoF

**Single-user (MIMO P2P):**

\[
\begin{array}{c}
\text{Tx} \quad \mathbf{H} \quad \text{Rx}
\end{array}
\]

**Capacity:**

\[
C = \log |\mathbf{I} + \mathbf{HH}^H|.
\]

**Optimization:** water-filling

**DoF:**

- \( M \) Tx antennas, \( N \) Rx antennas
  \[
  \Rightarrow \mathbf{H} \in \mathbb{C}^{N \times M} \Rightarrow \text{rank}(\mathbf{H}) = \min\{M, N\}
  \]

- \( C \approx \min\{M, N\}C(P) \) at high \( P \)
  \[
  C(P): \text{SISO P2P capacity}
  \]

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

Single-user (MIMO P2P):

\[ C = \log |I + HQH^H| \]

Optimization: water-filling

**DoF:**

- \( M \) Tx antennas, \( N \) Rx antennas
- \( H \in \mathbb{C}^{N \times M} \Rightarrow \text{rank}(H) = \min\{M, N\} \)
- \( C \approx \min\{M, N\}C(P) \) at high \( P \)
- **DoF:** \( d = \lim_{P \to \infty} \frac{\text{Capacity}}{C(P)} = \text{rank}(H) \Rightarrow d = \min\{M, N\} \)
- Capacity equivalent to that of \( d \) parallel SISO P2P channels!
Capacity to DoF

Multi-user (MIMO MAC):

\[
\begin{align*}
R_i & \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 & \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H|
\end{align*}
\]

Optimization: Iterative water-filling

- DoF:
  \(M_i\) Tx antennas, \(N\) Rx antennas
  \(\Rightarrow\) \(H_i \in \mathbb{C}^{N \times M_i}\)
  \(\Rightarrow\) \(\text{rank}(H_i) = \min\{M_i, N\}\)

- \(C_\Sigma \approx \min\{M_1 + M_2, N\}\)

DoF:

\[
d_i = \lim_{P \to \infty} R_i C(P)
\]

\(\Rightarrow\) DoF region:

\[
d_i \leq \text{rank}(H_i), \quad d_1 + d_2 \leq \text{rank}(\begin{bmatrix} H_1 & H_2 \end{bmatrix})
\]

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

\(\Rightarrow\) Sum-capacity equivalent to that of \(d_\Sigma\) parallel SISO P2P channels!
Capacity to DoF

Multi-user (MIMO MAC):

Capacity region:

\[ R_i \leq \log |I + H_i Q_i H_i^H| \]
\[ R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \]
Capacity to DoF

Multi-user (MIMO MAC):

```
Tx 2
   ▼ ▼
   H2
   ▼ ▼
Rx
   ▼ ▼
Tx 1
```

Capacity region:

\[
R_i \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H|
\]

Optimization: Iterative water-filling
Capacity to DoF

Multi-user (MIMO MAC):

\[
\begin{align*}
\text{Tx} 1 & \quad H_1 \\
\text{Tx} 2 & \quad H_2 \\
& \quad \text{Rx}
\end{align*}
\]

DoF:

- \( M_i \) Tx antennas, \( N \) Rx antennas

Capacity region:

\[
\begin{align*}
R_i & \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 & \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H|
\end{align*}
\]

Optimization: Iterative water-filling
Capacity to DoF

Multi-user (MIMO MAC):

\[ \text{Capacity region:} \]
\[ R_i \leq \log |I + H_i Q_i H_i^H| \]
\[ R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \]

Optimization: Iterative water-filling

DoF:

- \( M_i \) Tx antennas, \( N \) Rx antennas

\[ \Rightarrow H_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(H_i) = \min\{M_i, N\} \]
Capacity to DoF

Multi-user (MIMO MAC):

\[
\begin{align*}
\text{Rx} & \quad H_1 \quad H_2
\end{align*}
\]

\[\text{Tx 1} \quad \text{Tx 2}\]

Capacity region:

\[
R_i \leq \log |I + H_i Q_i H_i^H|
\]

\[
R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H|
\]

Optimization: Iterative water-filling

DoF:

- \(M_i\) Tx antennas, \(N\) Rx antennas

\[
\Rightarrow H_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(H_i) = \min\{M_i, N\}
\]

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

Anas Chaaban and Aydin Sezgin

Bi-Directional Communications
Capacity to DoF

Multi-user (MIMO MAC):

\[ \begin{align*}
\text{Tx 1} & \quad \begin{array}{c}
H_1 \\
\end{array} \\
\text{Tx 2} & \quad \begin{array}{c}
H_2
\end{array} \\
\end{align*} \Rightarrow \text{Rx} \]

Capacity region:

\[ \begin{align*}
R_i & \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 & \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H|
\end{align*} \]

Optimization: Iterative water-filling

DoF:

- \( M_i \) Tx antennas, \( N \) Rx antennas
- \( H_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(H_i) = \min\{M_i, N\} \)
- \( C_\Sigma \approx \min\{M_1 + M_2, N\} C(P) \) at high \( P \)
- \( \text{DoF: } d_i = \lim_{P \to \infty} \frac{R_i}{C(P)} \)
Capacity to DoF

Multi-user (MIMO MAC):

\[
\begin{align*}
\text{Tx 1} & \rightarrow H_1 \rightarrow \text{Rx} \\
\text{Tx 2} & \rightarrow H_2 \\
\end{align*}
\]

Capacity region:

\[
\begin{align*}
R_i & \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 & \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \\
\end{align*}
\]

Optimization: Iterative water-filling

DoF:

- \( M_i \) Tx antennas, \( N \) Rx antennas
- \( \mathbf{H}_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(\mathbf{H}_i) = \min\{M_i, N\} \)
- \( C_{\Sigma} \approx \min\{M_1 + M_2, N\} C(P) \) at high \( P \)
- DoF: \( d_i = \lim_{P \to \infty} \frac{R_i}{C(P)} \)

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

Multi-user (MIMO MAC):

\[ \begin{array}{c}
\text{Tx 1} \\
H_1 \\
\text{Rx} \\
H_2 \\
\text{Tx 2}
\end{array} \]

Capacity region:

\[ R_i \leq \log |I + H_i Q_i H_i^H| \]
\[ R_1 + R_2 \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \]

Optimization: Iterative water-filling

DoF:

- \( M_i \) Tx antennas, \( N \) Rx antennas
  \[ \Rightarrow H_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(H_i) = \min\{M_i, N\} \]
- \( C_\Sigma \approx \min\{M_1 + M_2, N\} C(P) \) at high \( P \)
- DoF: \( d_i = \lim_{P \to \infty} \frac{R_i}{C(P)} \)
  \[ \Rightarrow \text{DoF region: } d_i \leq \text{rank}(H_i), \quad d_1 + d_2 \leq \text{rank}([H_1, H_2]) \]
  \[ \Rightarrow d_\Sigma = \min\{M_1 + M_2, N\} \]
Capacity to DoF

Multi-user (MIMO MAC):

\[
\begin{align*}
\text{Tx 1} & \quad H_1 & \quad \text{Rx} \\
\text{Tx 2} & \quad H_2 \\
\end{align*}
\]

Capacity region:

\[
\begin{align*}
R_i & \leq \log |I + H_i Q_i H_i^H| \\
R_1 + R_2 & \leq \log |I + H_1 Q_1 H_1^H + H_2 Q_2 H_2^H| \\
\end{align*}
\]

Optimization: Iterative water-filling

DoF:

- \(M_i\) Tx antennas, \(N\) Rx antennas
  \[
  H_i \in \mathbb{C}^{N \times M_i} \Rightarrow \text{rank}(H_i) = \min\{M_i, N\}
  \]
- \(C_\Sigma \approx \min\{M_1 + M_2, N\} C(P)\) at high \(P\)
- DoF: \(d_i = \lim_{P \to \infty} \frac{R_i}{C(P)}\)
  \[
  H_1, H_2
  \]
  \[
  H_1, H_2
  \]
  \[
  H_1, H_2
  \]
  \[
  H_1, H_2
  \]
- DoF region: \(d_1 \leq \text{rank}(H_i), \quad d_1 + d_2 \leq \text{rank}([H_1, H_2])\)
- \(d_\Sigma = \min\{M_1 + M_2, N\}\)
- Sum-capacity equivalent to that of \(d_\Sigma\) parallel SISO P2P channels!
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

$$C_{\Sigma} = dC(P) + o(C(P)),$$

($C(P)$: capacity of a P2P 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
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),
⇒ 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 diagonalization)

Transform an arbitrary MIMO channel matrix $H$ to a diagonal matrix.
Channel Diagonalization

Definition (Channel diagonalization)

Transform an arbitrary MIMO channel matrix $H$ to a diagonal matrix.

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

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

\[
\begin{align*}
\mathbf{T}_x & \quad \mathbf{H} \\
\mathbf{x} & \quad \mathbf{y}
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{H}^\dagger & \quad \mathbf{H}^\dagger \mathbf{N} \\
\mathbf{x} & \quad \mathbf{y}
\end{align*}
\]
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:
  
  $$
  \mathbf{H}^\dagger = \mathbf{H}^H [\mathbf{H} \mathbf{H}^H]^{-1}
  $$

**Definition (Channel diagonalization)**

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

Definition (Channel diagonalization)
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:
  $\mathbf{H}^\ddagger = [\mathbf{H}^H \mathbf{H}^H]^{-1} \mathbf{H}^H$
- $\mathbf{H}^\ddagger \mathbf{H} = \mathbf{I}$

DoF achievable by treating each sub-channel separately $\Rightarrow$ Separability!
MAC and BC are also separable
Channel Diagonalization

Definition (Channel diagonalization)

Transform an arbitrary MIMO channel matrix $H$ 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:
  $$H^\dagger = H^H [HH^H]^{-1}$$
- $H^\dagger H = I$

$N$ parallel SISO P2P channels!
Channel Diagonalization

Definition (Channel diagonalization)

Transform an arbitrary MIMO channel matrix $H$ 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:
  $H^\dagger = H^H [HH^H]^{-1}$

- $H^\dagger H = I$

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

$N$ parallel SISO P2P channels!
Channel Diagonalization

**Definition (Channel diagonalization)**

Transform an arbitrary MIMO channel matrix $H$ 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:
  $$H^\dagger = H^H [HH^H]^{-1}$$
- $H^\dagger H = I$

$M \leq N$: ZF post-coding:
- Pseudo-inverse:
  $$H^\ddagger = [H^H H]^{-1} H^H$$

$N$ parallel SISO P2P channels!

DoF achievable by treating each sub-channel separately $\Rightarrow$ **Separability**! MAC and BC are also separable
Channel Diagonalization

Definition (Channel diagonalization)

Transform an arbitrary MIMO channel matrix $H$ 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:
  \[ H^\dagger = H^H [HH^H]^{-1} \]
- $H^\dagger H = I$

$M \leq N$: ZF post-coding:
- Pseudo-inverse:
  \[ H^{\dagger} = [H^H H]^{-1} H^H \]
- $HH^{\dagger} = I$

$N$ parallel SISO P2P channels!
Channel Diagonalization

Definition (Channel diagonalization)

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!

$M$ parallel SISO P2P channels!
Channel Diagonalization

Definition (Channel diagonalization)

Transform an arbitrary MIMO channel matrix $H$ 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:
  $$H^\dagger = H^H [HH^H]^{-1}$$
- $H^\dagger H = I$

$M \leq N$: ZF post-coding:
- Pseudo-inverse:
  $$H^\ddagger = [H^H H]^{-1} H^H$$
- $HH^\ddagger = I$

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

$N$ parallel SISO P2P channels!

$M$ parallel SISO P2P channels!
Channel Diagonalization

**Definition (Channel diagonalization)**

Transform an arbitrary MIMO channel matrix $H$ 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:
  $$H^\dagger = H^H [HH^H]^{-1}$$
- $H^\dagger H = I$

$M \leq N$: ZF post-coding:
- Pseudo-inverse:
  $$H^\ddagger = [H^H H]^{-1} H^H$$
- $HH^\ddagger = I$

DoF achievable by treating each sub-channel separately $\Rightarrow$ Separability!
MAC and BC are also separable
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
Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$. 

Two signals can be aligned by pre-coding:

$$x_1 = V_1 u_1$$
$$x_2 = V_2 u_2$$

$$V_1 H_1 = H_2 V_2$$

Useful for CF!
Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

$x_1 = V_1 u_1$
$x_2 = V_2 u_2$
Signal Alignment

Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

$x_1 = V_1 u_1$
$x_2 = V_2 u_2$

- $V_1$, $V_2$ arbitrary
Signal Alignment

Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

$x_1 = V_1 u_1$
$x_2 = V_2 u_2$

- $V_1, V_2$ arbitrary
Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

- $x_1 = V_1 u_1$
- $x_2 = V_2 u_2$

- $V_1$, $V_2$ arbitrary
- $H_1 V_1 = H_2 V_2$
Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

- $x_1 = V_1 u_1$
- $x_2 = V_2 u_2$

- $V_1, V_2$ arbitrary
- $H_1 V_1 = H_2 V_2$
**Signal Alignment**

### Definition (Signal alignment)

Placing two signals $x_1$ and $x_2$ in signal space so that $\text{span}(x_1) = \text{span}(x_2)$.

Two signals can be aligned by pre-coding:

- $x_1 = V_1 u_1$
- $x_2 = V_2 u_2$

- $V_1, V_2$ arbitrary
- $H_1 V_1 = H_2 V_2$
- Useful for CF!
Signal-alignment for CF

Definition (Compute-forward (CF))

Computing and forwarding a linear combination of two signals e.g. $x_1 + x_2$. 
### Signal-alignment for CF

**Definition (Compute-forward (CF))**

Computing and forwarding a linear combination of two signals e.g. $x_1 + x_2$.

**Recall:** CF can be accomplished by using lattice codes.

![Diagram of nodes and signal points](image-url)
Signal-alignment for CF

**Definition (Compute-forward (CF))**

Computing and forwarding a linear combination of two signals e.g. \( x_1 + x_2 \).

Recall: CF can be accomplished by using lattice codes.

**Example:**
- \( x_1 = V_1 u_1 \),
- \( x_2 = V_2 u_2 \),
- \( H_1 V_1 = H_2 V_2 = [1, 1]^T \)
**Signal-alignment for CF**

**Definition (Compute-forward (CF))**

Computing and forwarding a linear combination of two signals e.g. $x_1 + x_2$. 

Recall: CF can be accomplished by using lattice codes.

**Example:**

- $x_1 = V_1u_1$,  
  $x_2 = V_2u_2$,  
  $H_1V_1 = H_2V_2 = [1, 1]^T$
- $y_3 = (u_1 + u_2)[1, 1]^T + n$
Signal-alignment for CF

**Definition (Compute-forward (CF))**

Computing and forwarding a linear combination of two signals e.g. \(x_1 + x_2\).

Recall: CF can be accomplished by using lattice codes.

Example:

- \(x_1 = V_1 u_1\),
- \(x_2 = V_2 u_2\),
- \(H_1 V_1 = H_2 V_2 = [1, 1]^T\)
- \(y_3 = (u_1 + u_2)[1, 1]^T + n\)
- compute \(u_1 + u_2\) from \([1, 1]y_3 = 2(u_1 + u_2) + n'\)
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:

\[
\begin{align*}
\mathbf{y}_r &= \mathbf{H}_1 \mathbf{x}_1 + \mathbf{H}_2 \mathbf{x}_2 + \mathbf{z}_r, \\
\mathbf{y}_i &= \mathbf{D}_i \mathbf{x}_r + \mathbf{z}_i,
\end{align*}
\]

\( \mathbf{H}_i, \mathbf{D}_i: m \times m \).
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:
  \[
  y_r = H_1 x_1 + H_2 x_2 + z_r, \quad y_i = D_i x_r + z_i, \quad H_i, D_i: m \times m.
  \]
- Channel diagonalization: \( x_i = H_i^{-1} u_i, y'_i = D_i^{-1} y_i \):
  \[
  y_r = u_1 + u_2 + z_r, \quad y'_i = x_r + z'_i.
  \]
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:
  \[
  y_r = H_1 x_1 + H_2 x_2 + z_r, \quad y_i = D_i x_r + z_i, \quad H_i, D_i: m \times m.
  \]
- Channel diagonalization: \( x_i = H_i^{-1} u_i, y_i' = D_i^{-1} y_i \):
  \[
  y_r = u_1 + u_2 + z_r, \quad y_i' = x_r + z_i',
  \]
  \( m \) 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:

\[
\begin{align*}
\mathbf{y}_r &= \mathbf{H}_1 \mathbf{x}_1 + \mathbf{H}_2 \mathbf{x}_2 + \mathbf{z}_r, \\
\mathbf{y}_i &= \mathbf{D}_i \mathbf{x}_r + \mathbf{z}_i, \\
\mathbf{H}_i, \mathbf{D}_i &\text{: } m \times m.
\end{align*}
\]

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

\[
\begin{align*}
\mathbf{y}_r &= \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{z}_r, \\
\mathbf{y}'_i &= \mathbf{x}_r + \mathbf{z}'_i,
\end{align*}
\]

\( \Rightarrow \) \( m \) parallel SISO two-way relay channels

- Apply CF: achieve

\[
2 \left[ \frac{1}{2} \log \left( \frac{1}{2} + P \right) \right]^+ \approx 2C(P) \text{ (at high } P\text{) per sub-channel}.
\]

Anas Chaaban and Aydin Sezgin
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:
  \[
y_r = H_1 x_1 + H_2 x_2 + z_r, \quad y_i = D_i x_r + z_i, \quad H_i, D_i: m \times m.
\]
- Channel diagonalization: \( x_i = H_i^{-1} u_i, y_i' = D_i^{-1} y_i \):
  \[
y_r = u_1 + u_2 + z_r, \quad y_i' = x_r + z_i',
\]
  \[
  m \text{ parallel SISO two-way relay channels}
\]
- 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,
⇒ 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 Y-channel [Lee & Lim 09]

- inputs $x_i$ and $x_r$,
- outputs
  
  \[ y_r = \sum_{i=1}^{K} H_i x_i + z_r \]
  and
  
  \[ y_i = D_i x_r + z_i, \]
MIMO Multi-way Relay Channel

MIMO Y-channel [Lee & Lim 09]

- inputs $x_i$ and $x_r$,
- outputs
  
  $y_r = \sum_{i=1}^{K} H_i x_i + z_r$ and
  
  $y_i = D_i x_r + z_i$,
- channels $H_i: N \times M_i$, and
  
  $D_i: M_i \times N$, 
MIMO Y-channel [Lee & Lim 09]

- inputs $x_i$ and $x_r$,
- outputs
  
  $$y_r = \sum_{i=1}^{K} H_i x_i + z_r$$
  
  and
  
  $$y_i = D_i x_r + z_i,$$
- channels $H_i: N \times M_i$, and
  $D_i: M_i \times N$,
- 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., $d = 3M$, 

\[ d = \min \{ M_1 + M_2 + M_3, 2M_2 + 2M_3, 2N \} \]
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

$$d = \min\{M_1 + M_2 + M_3, 2M_2 + 2M_3, 2N\}.$$
Transmission Strategy

Signal-space alignment for network-coding

- The signals $H_1x_1$, $H_2x_2$, and $H_3x_3$ fill the entire space at the relay
Transmission Strategy

Signal-space alignment for network-coding

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

Signal-space alignment for network-coding

- The signals $H_1 x_1$, $H_2 x_2$, and $H_3 x_3$ fill the entire space at the relay
- Relay zero-forces $H_2 x_2$ using $N_{13}$ and $H_3 x_3$ using $N_{12}$
- Result: $N_{12} H_1 x_1 + N_{12} H_2 x_2$ (2D) and $N_{13} H_1 x_1 + N_{13} H_3 x_3$ (1D)
Transmission Strategy

Signal-space alignment for network-coding

- The signals $H_1 x_1$, $H_2 x_2$, and $H_3 x_3$ fill the entire space at the relay
- Relay zero-forces $H_2 x_2$ using $N_{13}$ and $H_3 x_3$ using $N_{12}$
- Result: $N_{12} H_1 x_1 + N_{12} H_2 x_2$ (2D) and $N_{13} H_1 x_1 + N_{13} H_3 x_3$ (1D)
- Tx's diagonalize their effective channels
The signals $H_1 x_1$, $H_2 x_2$, and $H_3 x_3$ fill the entire space at the relay.

Relay zero-forces $H_2 x_2$ using $N_{13}$ and $H_3 x_3$ using $N_{12}$.

Result: $N_{12} H_1 x_1 + N_{12} H_2 x_2$ (2D) and $N_{13} H_1 x_1 + N_{13} H_3 x_3$ (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

- 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)
Transmission strategy

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

\[
\begin{bmatrix}
T_{12} & T_{13}
\end{bmatrix}
\]
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)
**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 \(u_{12}\) and \(u_{21}\) (\(M_2\)-dim), and \(u_{13}\) and \(u_{31}\) (\(M_3\)-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 \( u_{12} \) and \( u_{21} \) (\( M_2 \)-dim), and \( u_{13} \) and \( u_{31} \) (\( M_3 \)-dim)
- Align \( u_{12} \) and \( u_{21} \), and align \( u_{13} \) and \( u_{31} \) @ relay
- Relay decodes \( L(u_{12}, u_{21}) \) and \( L(u_{13}, u_{31}) \)
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
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
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
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 $u_{12}$ and $u_{21}$ in a $d_{12}$-dim subspace
  
  \[
  d_{12} = \dim(\text{span}(H_1) \cap \text{span}(H_2)) = \frac{M_1 + M_2 - M_3}{2}
  \]
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 \( u_{12} \) and \( u_{21} \) in a \( d_{12} \)-dim subspace
  \( (d_{12} = \dim(\text{span}(H_1) \cap \text{span}(H_2)) = \frac{M_1 + M_2 - M_3}{2}) \)
- Similarly, align \( u_{13} \) and \( u_{31} \) in \( d_{13} = \frac{M_1 + M_3 - M_2}{2} \) dimensions, and align \( u_{23} \) and \( u_{32} \) in \( d_{23} = \frac{M_2 + M_3 - M_1}{2} \) dimensions
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
  \( (d_{12} = \dim(\text{span}(\mathbf{H}_1) \cap \text{span}(\mathbf{H}_2)) = \frac{M_1 + M_2 - M_3}{2}) \)
- 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 $N = \min\{2M_2 + 2M_3, M_1 + M_2 + M_3\}$
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
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**

\[ R_{ij} : \text{Rate of signal from } i \text{ to } j \]
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

\[ R_{ij}: \text{Rate of signal from } i \text{ to } j \]
\[ d_{ij}: \text{DoF defined as } \lim_{P \to \infty} \frac{R_{ij}}{C(P)} \]
\[ d_\Sigma: \text{sum-DoF } = \sum d_{ij} \]
**Importance of DoF region**

**Definition**

- $R_{ij}$: Rate of signal from $i$ to $j$
- $d_{ij}$: DoF defined as $\lim_{P \to \infty} \frac{R_{ij}}{C(P)}$
- $d_\Sigma$: sum-DoF $= \sum d_{ij}$
- $\mathcal{D}$: Set of simultaneously achievable DoF’s $d_{ij}$

The diagram illustrates the relationship between $d_{13}$ and $d_{12}$.
## Importance of DoF region

### Definition

- $R_{ij}$: Rate of signal from $i$ to $j$
- $d_{ij}$: DoF defined as $\lim_{P \to \infty} \frac{R_{ij}}{C(P)}$
- $d_\Sigma$: sum-DoF = $\sum d_{ij}$
- $D$: Set of simultaneously achievable DoF’s $d_{ij}$

**Additional Information**

$d_\Sigma$ 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 $\lim_{P \to \infty} \frac{R_{ij}}{C(P)}$
- $d_{\Sigma}$: sum-DoF $= \sum d_{ij}$
- $D$: Set of simultaneously achievable DoF's $d_{ij}$

+ $d_{\Sigma}$ is an overall measure of performance for a network
- $d_{\Sigma}$ 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 $\lim_{P \to \infty} \frac{R_{ij}}{C(P)}$

$d_\Sigma$: sum-DoF $= \sum d_{ij}$

$D$: Set of simultaneously achievable DoF’s $d_{ij}$

+ $d_\Sigma$ is an overall measure of performance for a network

- $d_\Sigma$ 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 \( \lim_{P \to \infty} \frac{R_{ij}}{C(P)} \)

\( d_{\Sigma} \): sum-DoF = \( \sum d_{ij} \)

\( D \): Set of simultaneously achievable DoF’s \( d_{ij} \)

+ \( d_{\Sigma} \) is an overall measure of performance for a network

− \( d_{\Sigma} \) 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 $\lim_{P \to \infty} \frac{R_{ij}}{C(P)}$
- $d_{\Sigma}$: sum-DoF $= \sum d_{ij}$
- $D$: Set of simultaneously achievable DoF’s $d_{ij}$

+ $d_{\Sigma}$ is an overall measure of performance for a network
- $d_{\Sigma}$ 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
\( M = N = 3 \)

<table>
<thead>
<tr>
<th></th>
<th>( d_{12} )</th>
<th>( d_{13} )</th>
<th>( d_{21} )</th>
<th>( d_{23} )</th>
<th>( d_{31} )</th>
<th>( d_{32} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Simple example

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

Let's try uni-directional

<table>
<thead>
<tr>
<th>$d_{12}$</th>
<th>$d_{13}$</th>
<th>$d_{21}$</th>
<th>$d_{23}$</th>
<th>$d_{31}$</th>
<th>$d_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Anas Chaaban and Aydin Sezgin
Simple example

**Goal:** Achieve the DoF tuple:
over a Y-channel with
\( M = N = 3 \)

Lets try uni-directional

<table>
<thead>
<tr>
<th>( d_{12} )</th>
<th>( d_{13} )</th>
<th>( d_{21} )</th>
<th>( d_{23} )</th>
<th>( d_{31} )</th>
<th>( d_{32} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Simple example

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

Let's try uni-directional

\[
\begin{array}{cccccc}
    d_{12} & d_{13} & d_{21} & d_{23} & d_{31} & d_{32} \\
    2 & 0 & 1 & 1 & 1 & 0 \\
\end{array}
\]
Simple example

**Goal:** Achieve the DoF tuple: over a Y-channel with

\[ M = N = 3 \]

Lets try uni-directional

\[ u_{12} \]

\[ v_{23} \]

\[ v_{31} \]

\[ d_{12} \quad d_{13} \quad d_{21} \quad d_{23} \quad d_{31} \quad d_{32} \]

\[ \begin{array}{cccccc}
2 & 0 & 1 & 1 & 1 & 0
\end{array} \]
Simple example

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

Lets try uni-directional

<table>
<thead>
<tr>
<th>$d_{12}$</th>
<th>$d_{13}$</th>
<th>$d_{21}$</th>
<th>$d_{23}$</th>
<th>$d_{31}$</th>
<th>$d_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Anas Chaaban and Aydin Sezgin

Bi-Directional Communications
Simple example

**Goal:** Achieve the DoF tuple:
over a Y-channel with
\( M = N = 3 \)

Let's try uni-directional + bi-directional

\[
\begin{bmatrix}
\text{doFs} \\
\text{User 1} \\
\text{User 2} \\
\text{User 3} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{doFs} \\
\text{Relay} \\
\text{User 1} \\
\text{User 2} \\
\text{User 3} \\
\end{bmatrix}
\]

\[
\begin{array}{cccccc}
\text{d}_{12} & \text{d}_{13} & \text{d}_{21} & \text{d}_{23} & \text{d}_{31} & \text{d}_{32} \\
2 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]
Simple example

**Goal:** Achieve the DoF tuple: over a Y-channel with \( M = N = 3 \)

<table>
<thead>
<tr>
<th>( d_{12} )</th>
<th>( d_{13} )</th>
<th>( d_{21} )</th>
<th>( d_{23} )</th>
<th>( d_{31} )</th>
<th>( d_{32} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( \lambda )</td>
<td>( \lambda )</td>
</tr>
</tbody>
</table>

Let's try uni-directional + bi-directional
Simple example

**Goal:** Achieve the DoF tuple: over a Y-channel with \( M = N = 3 \)

Let's try uni-directional + bi-directional
Simple example

**Goal:** Achieve the DoF tuple: over a Y-channel with \( M = N = 3 \)

Let's try uni-directional + bi-directional
Simple example

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

\[
\begin{array}{ccccccc}
    d_{12} & d_{13} & d_{21} & d_{23} & d_{31} & d_{32} \\
    2 & 1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Let's try uni-directional + bi-directional
Simple example

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

<table>
<thead>
<tr>
<th></th>
<th>$d_{12}$</th>
<th>$d_{13}$</th>
<th>$d_{21}$</th>
<th>$d_{23}$</th>
<th>$d_{31}$</th>
<th>$d_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Let's try uni-directional + bi-directional + Cyclic
Simple example

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

 Lets try uni-directional + bi-directional + Cyclic

<table>
<thead>
<tr>
<th>$d_{12}$</th>
<th>$d_{13}$</th>
<th>$d_{21}$</th>
<th>$d_{23}$</th>
<th>$d_{31}$</th>
<th>$d_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Anas Chaaban and Aydin Sezgin

Bi-Directional Communications
**Simple example**

**Goal:** Achieve the DoF tuple: 
over a Y-channel with 
\[ M = N = 3 \]

Let's try uni-directional + bi-directional + Cyclic

<table>
<thead>
<tr>
<th></th>
<th>( d_{12} )</th>
<th>( d_{13} )</th>
<th>( d_{21} )</th>
<th>( d_{23} )</th>
<th>( d_{31} )</th>
<th>( d_{32} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Simple example

**Goal:** Achieve the DoF tuple:
over a Y-channel with
\( M = N = 3 \)

Let's try uni-directional + bi-directional + Cyclic

\[
\begin{array}{cccccccc}
& d_{12} & d_{13} & d_{21} & d_{23} & d_{31} & d_{32} \\
\hline
2 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
\]
Simple example

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

<table>
<thead>
<tr>
<th>$d_{12}$</th>
<th>$d_{13}$</th>
<th>$d_{21}$</th>
<th>$d_{23}$</th>
<th>$d_{31}$</th>
<th>$d_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Let's try uni-directional + bi-directional + Cyclic

Anas Chaaban and Aydin Sezgin
Bi-Directional Communications
Simple example

**Goal:** Achieve the DoF tuple: over a Y-channel with
\[ M = N = 3 \]

Let's try uni-directional + bi-directional + Cyclic ← optimal combination

<table>
<thead>
<tr>
<th>( d_{12} )</th>
<th>( d_{13} )</th>
<th>( d_{21} )</th>
<th>( d_{23} )</th>
<th>( d_{31} )</th>
<th>( d_{32} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$d_{12} + d_{13} + d_{23} \leq N$$
The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

\[ d_{12} + d_{13} + d_{23} \leq N \]
\[ d_{12} + d_{13} + d_{32} \leq N \]
The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$d_{12} + d_{13} + d_{23} \leq N$$

$$d_{12} + d_{13} + d_{32} \leq N$$

$$\vdots \quad \vdots \quad \vdots \quad \leq \ \vdots$$
The DoF region of a 3-user MIMO Y-channel with $N \leq M$ is described by

$$d_{12} + d_{13} + d_{23} \leq N$$
$$d_{12} + d_{13} + d_{32} \leq N$$
$$\vdots \quad \vdots \quad \vdots \quad \leq \vdots$$

**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.
Achievability of $\mathcal{D}$ is proved using:

- Channel diagonalization: MIMO $Y$-channel $\rightarrow$ 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
Overview

Achievability of $\mathcal{D}$ is proved using:

**Channel diagonalization:**

- MIMO
- Y-channel

![Diagram](attachment:image.png)
Achievability of $\mathcal{D}$ is proved using:

**Channel diagonalization:**
- MIMO $\rightarrow$ $N$ SISO
- Y-channel $\rightarrow$ 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** $\rightarrow$ $N$ **SISO**
- **Y-channel** $\rightarrow$ **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
Achievability of $D$ is proved using:

**Channel diagonalization:**
- MIMO $\rightarrow$ $N$ SISO
- Y-channel $\rightarrow$ Y-channels
  (sub-channels)

**Information exchange:**
- Bi-directional: signal-alignment/compute-forward
Achievability of $\mathcal{D}$ is proved using:

**Channel diagonalization:**
- MIMO $\rightarrow$ $N$ SISO
- Y-channel $\rightarrow$ Y-channels (sub-channels)

**Information exchange:**
- Bi-directional: signal-alignment/compute-forward
- Cyclic: signal-alignment/compute-forward
Achievability of $\mathcal{D}$ is proved using:

**Channel diagonalization:**
- MIMO → $N$ SISO
- $Y$-channel → $Y$-channels (sub-channels)

**Information exchange:**
- Bi-directional: signal-alignment/compute-forward
- Cyclic: signal-alignment/compute-forward
- Uni-directional: decode-forward
Achievability of $D$ is proved using:

**Channel diagonalization:**

MIMO $\rightarrow$ $N$ SISO

Y-channel $\rightarrow$ 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
• a MIMO Y-channel with $M = N = 3$

Channel Diagonalization

User 1
\[ x_1 \]

User 2
\[ x_2 \]

Relay
\[ H_1 \]

User 3
\[ x_3 \]

User 1
\[ D_1 \]

User 2
\[ D_2 \]

Relay
\[ H_2 \]

User 3
\[ D_3 \]

User 1
\[ y_1 \]

User 2
\[ y_2 \]

Relay
\[ x_r \]

User 3
\[ y_3 \]
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

![Bi-directional Communication Diagram](image_url)
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 $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 symbols</td>
<td>1 sub-channel</td>
<td>2 sub-channels</td>
<td>1 sub-channel</td>
</tr>
<tr>
<td>3 symbols</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 symbol</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:**

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>
Back to our example

DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

**Uni-directional only:**
- 5 sub-channels $> N!$

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbols</th>
<th>Sub-channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-directional</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Cyclic</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Back to our example

DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

**Uni-directional only:**
- 5 sub-channels $> N!$

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td></td>
<td>3 symbols</td>
<td>2 sub-channels</td>
</tr>
<tr>
<td>Uni-directional</td>
<td></td>
<td>1 symbol</td>
<td>1 sub-channel</td>
</tr>
</tbody>
</table>

**Bi-directional + uni-directional:**

- bi-directional achieves $d_{b12} = d_{b21} = 1$ 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 \( 3 = N \leq M \)

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

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>
Back to our example

DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, $Y$-channel with $3 = N \leq M$

Uni-directional only:

- 5 sub-channels $> N!$

Bi-directional + uni-directional:

- bi-directional achieves $d^b_{12} = d^b_{21} = 1$ over 1 sub-channel
- residual DoF $(1, 0, 0, 1, 1, 0)$
Back to our example

DoF tuple \( d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0) \), Y-channel with \( 3 = N \leq M \)

**Uni-directional only:**
- 5 sub-channels > \( N! \)

**Bi-directional + uni-directional:**
- bi-directional achieves \( d^b_{12} = d^b_{21} = 1 \) over 1 sub-channel
- residual DoF \((1, 0, 0, 1, 1, 0)\)
- uni-directional needs 3 more sub-channels

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>

Back to our example

DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

**Uni-directional only:**
- 5 sub-channels $> N$!

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>

**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 $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

Uni-directional only:

- 5 sub-channels $> N!$

<table>
<thead>
<tr>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
</tr>
</tbody>
</table>

Bi-directional + uni-directional:

- bi-directional achieves $d^b_{12} = d^b_{21} = 1$ 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 \( 3 = N \leq M \)

### Uni-directional only:

- 5 sub-channels > \( N! \)

### Bi-directional + uni-directional:

- bi-directional achieves \( d^b_{12} = d^b_{21} = 1 \) 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 \( d^b_{12} = d^b_{21} = 1 \) over 1 sub-channel

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>
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 + uni-directional:**
- bi-directional achieves \( d^b_{12} = d^b_{21} = 1 \) 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 \( > N! \)

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbols</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>sub-channels</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Bi-directional + cyclic + uni-directional:**
- bi-directional achieves \( d^b_{12} = d^b_{21} = 1 \) over 1 sub-channel
- residual DoF \( (1, 0, 0, 1, 1, 0) \)
Back to our example

DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0)$, Y-channel with $3 = N \leq M$

**Uni-directional only:**
- 5 sub-channels $> N!$

<table>
<thead>
<tr>
<th>Method</th>
<th>Symbols</th>
<th>Sub-channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-directional</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Cyclic</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**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 \( d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0) \), Y-channel with \( 3 = N \leq M \)

**Uni-directional only:**
- 5 sub-channels > \( N! \)

**Bi-directional + 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 \( d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) = (2, 0, 1, 1, 1, 0) \), Y-channel with \( 3 = N \leq M \)

**Uni-directional only:**
- 5 sub-channels > \( N! \)

**Bi-directional + uni-directional:**
- bi-directional achieves \( d^b_{12} = d^b_{21} = 1 \) 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 \( d^b_{12} = d^b_{21} = 1 \) over 1 sub-channel
- residual DoF \((1, 0, 0, 1, 1, 0)\)
- cyclic achieves \( d^c_{12} = d^c_{23} = d^c_{31} = 1 \) over 2 sub-channels
- residual DoF \((0, 0, 0, 0, 0, 0)\)
- total number of sub-channels \( 3 = N! \)
Back to our example

User 1

User 2

User 3

Relay

H1

H2

H3

D1

D2

D3

u12

v12

u21

v23

v31

u12 + u21

v23 + v31

Anas Chaaban and Aydin Sezgin

Bi-Directional Communications
In General

Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \)
Consider a DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2$-cycles and 3-cycles!
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:**

![Diagram showing bi-directional communication with nodes 1, 2, and 3 connected by edges labeled with $d_{ij}$ values.]
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:** resolves 2-cycles

Bi-directional communications
In General

Consider a DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow$ 2-cycles and 3-cycles!

**Bi-directional:**
resolves 2-cycles

Residual DoF tuple (e.g.) $d' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow$ 3-cycle!
In General

Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**Bi-directional:**
resolves 2-cycles

Residual DoF tuple (e.g.) \( \mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow 3\text{-cycle}! \)

**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
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**Bi-directional:**
resolves 2-cycles

Residual DoF tuple (e.g.) \( \mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow 3\text{-cycle}! \)

**Cyclic:**
resolves 3-cycles
Residual DoF tuple (e.g.) \( \mathbf{d}' = (d''_{12}, 0, 0, d''_{23}, 0, 0) \Rightarrow \text{no cycles}! \)
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:**
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**Bi-directional:**
resolves 2-cycles

Residual DoF tuple (e.g.) \( \mathbf{d}' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \Rightarrow 3\text{-cycle}! \)

**Cyclic:**
resolves 3-cycles
Residual DoF tuple (e.g.) \( \mathbf{d}' = (d''_{12}, 0, 0, d''_{23}, 0, 0) \Rightarrow \text{no cycles}! \)

**Uni-directional:**
\( \mathbf{d} \text{ achieved!} \)
For the $K$-user Y-channel with $N \leq M$:

- $2$-cycles up to $K$-cycles,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
- exchanges $\ell$ symbols
- requires $\ell - 1$ dimensions
- efficiency $\frac{\ell}{\ell - 1}$

$\text{DoF region described by}$

$$K - 1 \sum_{i=1}^{K} \sum_{j=i+1}^{K} d_{p_i} p_j \leq N,$$

where $p$ is a permutation of $(1, 2, \ldots, K)$. 

Cyclic communication requires joint encoding over multiple sub-channels $\Rightarrow$ MIMO Y-channels are in general inseparable!
For the $K$-user Y-channel with $N \leq M$:

- 2-cycles up to $K$-cycles,
For the $K$-user Y-channel with $N \leq M$:

- 2-cycles up to $K$-cycles,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
For the $K$-user Y-channel with $N \leq M$:

- 2-cycles up to $K$-cycles,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
- exchanges $\ell$ symbols

Cyclic communication requires joint encoding over multiple sub-channels $\Rightarrow$ MIMO Y-channels are in general inseparable!
For the $K$-user Y-channel with $N \leq M$:

- 2-cycles up to $K$-cycles,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
- exchanges $\ell$ symbols
- requires $\ell - 1$ dimensions
For the $K$-user Y-channel with $N \leq M$:

- 2-cycles up to $K$-cycles,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
- exchanges $\ell$ 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,
- $\ell$-cycles resolved by an $\ell$-cyclic strategy
- exchanges $\ell$ symbols
- requires $\ell - 1$ dimensions
- efficiency $\ell/(\ell - 1)$

• DoF region described by

\[
\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d_{p_ip_j} \leq N, \quad \forall p
\]

where $p$ is a permutation of $(1, 2, \cdots, K)$. 
**For the \( K \)-user Y-channel with \( N \leq M \):**

- 2-cycles up to \( K \)-cycles,
- \( \ell \)-cycles resolved by an \( \ell \)-cyclic strategy
- exchanges \( \ell \) symbols
- requires \( \ell - 1 \) dimensions
- efficiency \( \ell / (\ell - 1) \)

- DoF region described by

\[
\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d_{p_ip_j} \leq N, \quad \forall p
\]

where \( p \) is a permutation of \((1, 2, \cdots, K)\).

Cyclic communication requires joint encoding over multiple sub-channels \( \Rightarrow \)
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

\[ y_k = H_{kj} x_j + H_{ki} x_i + z_k, \]

- \( y_k \) represents the received signal at node \( k \).
- \( H_{kj} \) is the channel matrix from node \( j \) to node \( k \).
- \( x_j \) is the transmitted signal from node \( j \).
- \( x_i \) is the transmitted signal from node \( i \).
- \( z_k \) is the noise at node \( k \).
\[ y_k = H_{kj} x_j + H_{ki} x_i + z_k, \]

- Capacity scaling (DoF)?
Transmission Scheme

Communication $i \leftrightarrow j$:
- Node $i$ sends $x_i = V_{ji} u_i$, node $j$ sends $x_j = V_{ij} u_j$
**Transmission Scheme**

Communication $i \leftrightarrow j$:

- Node $i$ sends $x_i = V_{ji}u_i$, node $j$ sends $x_j = V_{ij}u_j$
- Alignment at node $k$: Set $\text{span}(H_{ki}V_{ji}) = \text{span}(H_{kj}V_{ij})$
Transmission Scheme

**Communication** $i \leftrightarrow j$:

- Node $i$ sends $x_i = V_{ji} u_i$, node $j$ sends $x_j = V_{ij} u_j$
- Alignment at node $k$: Set $\text{span}(H_{ki} V_{ji}) = \text{span}(H_{kj} V_{ij})$
Achievable DoF:

- If $M_1 \geq M_2 \geq M_3$, DoF is $2M_2$
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:

Multi-way Communications

Anas Chaaban
and
Aydin Sezgin
All this and more to appear in:

Thank you for your attention!
Optimality

<table>
<thead>
<tr>
<th></th>
<th>Symbols</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-directional</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Cyclic</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

No cycles $\Rightarrow N_{s} \leq N \Rightarrow$ All $d \in D$ are achievable
**Optimality**

Total number of dimensions required to achieve \( \mathbf{d} \in \mathcal{D} \):

\[
N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{i,j}^b + \sum_{j=2}^{3} 2d_{1,j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{i,j}^u
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbols</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-directional</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Cyclic</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

No cycles \( \Rightarrow N_s \leq N \Rightarrow \) All \( d \in \mathcal{D} \) are achievable.
Optimality

Total number of dimensions required to achieve $d \in D$:

\[
N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{i,j}^b + \sum_{j=2}^{3} 2d_{1,j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{i,j}^u
\]

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Symbols</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Number of Sub-channels</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

\[d_{i,j}^u = d_{i,j} - d_{i,j}^b - d_{i,j}^c\]
# Optimality

Total number of dimensions required to achieve \( d \in D \): 

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 symbols</td>
<td>3 symbols</td>
<td>1 symbol</td>
</tr>
<tr>
<td></td>
<td>1 sub-channel</td>
<td>2 sub-channels</td>
<td>1 sub-channel</td>
</tr>
</tbody>
</table>

\[
N_s = \sum_{i=1}^{3} \sum_{j=i+1}^{3} b_{ij} + \sum_{j=2}^{3} 2c_{1j} + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} u_{ij} \\
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} b_{ij} - \sum_{j=2}^{3} d_{1j}^c \\
(d_{ij}^u = d_{ij} - b_{ij} - c_{ij})
\]

No cycles \( \Rightarrow N_s \leq N \Rightarrow \) All \( d \in D \) are achievable
Optimality

Total number of dimensions required to achieve \( d \in D \):

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 symbols</td>
<td>3 symbols</td>
<td>1 symbol</td>
</tr>
<tr>
<td></td>
<td>1 sub-channel</td>
<td>2 sub-channels</td>
<td>1 sub-channel</td>
</tr>
</tbody>
</table>

\[
N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b + \sum_{i=1}^{3} 2d_{1j}^c + \sum_{i=1}^{3} \sum_{j=1, j\neq i}^{3} d_{ij}^u
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j\neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c
\]

\( (d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c) \)

\( (d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\}) \)
Optimality

Total number of dimensions required to achieve \( d \in \mathcal{D} \):

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>

\[
N_s = \begin{cases} 
_{\text{bi-directional}} \sum_{i=1}^{2} \sum_{j=i+1}^{3} d^b_{ij} + \sum_{i=1}^{3} 2d^c_{1j} + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d^u_{ij} & (d^u_{ij} = d_{ij} - d^b_{ij} - d^c_{ij}) \\
_{\text{cyclic}} \sum_{i=1}^{3} \sum_{j=1}^{3} d_{ij} - \sum_{i=1}^{2} \sum_{j=i+1}^{3} d^b_{ij} - \sum_{j=2}^{3} d^c_{1j} & (d_{ij} + d_{ji} - d^b_{ij} = \max\{d_{ij}, d_{ji}\}) \\
_{\text{uni-directional}} \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d^c_{12} - d^c_{13} & d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d^c_{13} = \max\{d_{12}, d_{21}\}
\end{cases}
\]

\[
d^b_{12} = d_{21}
\]
Optimality

Total number of dimensions required to achieve $d \in D$:

\[
N_s = \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b + \sum_{i=1}^{3} \sum_{j=2}^{3} 2d_{1j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij}^u
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c
\]

\[
= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c
\]

\[
= d_{12} + d_{23} + d_{31} - d_{12}^c
\]

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 symbols</td>
<td>3 symbols</td>
<td>1 symbol</td>
</tr>
<tr>
<td></td>
<td>1 sub-channel</td>
<td>2 sub-channels</td>
<td>1 sub-channel</td>
</tr>
</tbody>
</table>

\[
d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c
\]

\[
d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\}
\]

\[
d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \ d_{12}^b = d_{21}
\]

\[
= d_{12} + d_{23} + d_{31} - d_{12}^c
\]
Optimality

Total number of dimensions required to achieve \( d \in \mathcal{D} \): 

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 symbols</td>
<td>1 sub-channel</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
N_s = \begin{cases} 
\sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b & \text{bi-directional} \\
\sum_{j=2}^{3} 2d_{1j}^c & \text{cyclic} \\
\sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij}^u & \text{uni-directional} 
\end{cases} 
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c 
\]

\[
= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c 
\]

\[
d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \quad d_{12}^b = d_{21} 
\]

\[
= d_{12} + d_{23} + d_{31} - d_{12}^c 
\]

\[
(d_{12}^c = d_{12} - d_{12}^b \text{ e.g.}) 
\]
Optimality

Total number of dimensions required to achieve $d \in \mathcal{D}$:

\[
N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b + \sum_{j=2}^{3} 2d_{1j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij}^u
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c
\]

\[
= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c
\]

\[
d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \quad d_{12}^b = d_{21}
\]

\[
= d_{12} + d_{23} + d_{31} - d_{12}^c
\]

\[
= d_{12}^b + d_{23} + d_{31}
\]

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>2 symbols</th>
<th>1 sub-channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>3 symbols</td>
<td>2 sub-channels</td>
<td></td>
</tr>
<tr>
<td>Uni-directional</td>
<td>1 symbol</td>
<td>1 sub-channel</td>
<td></td>
</tr>
</tbody>
</table>

\(d_{ij}^u = d_{ij} - d_{ij}^b - d_{ij}^c\)
Optimality

Total number of dimensions required to achieve $d \in D$:

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Sub-channels</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$$N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{i,j}^b + \sum_{j=2}^{3} 2d_{1,j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{i,j}^u$$

$$= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{i,j} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{i,j}^b - \sum_{j=2}^{3} d_{1,j}^c$$

$$= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c$$

$$d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \quad d_{12}^b = d_{21}$$

$$= d_{12} + d_{23} + d_{31} - d_{12}^c$$

$$= d_{12}^b + d_{23} + d_{31}$$

$$= d_{21} + d_{23} + d_{31}$$

$$d_{12}^c = d_{12} - d_{12}^b \text{ e.g.}$$

$$\Rightarrow d_{13}^c = 0, \quad d_{12}^b = d_{21}$$
Optimality

Total number of dimensions required to achieve \( \mathbf{d} \in \mathcal{D} \):

\[
N_s = \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b + \sum_{i=1}^{3} 2d_{1j}^c + \sum_{i=1}^{3} \sum_{j=2}^{3} d_{ij}^u
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c
\]

\[
= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c
\]

\[
d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \ d_{12}^b = d_{21}
\]

\[
= d_{12} + d_{23} + d_{31} - d_{12}^c
\]

\[
= d_{12}^b + d_{23} + d_{31}
\]

\[
= d_{21} + d_{23} + d_{31}
\]

No cycles \( \Rightarrow N_s \leq N \)

<table>
<thead>
<tr>
<th></th>
<th>Bi-directional</th>
<th>Cyclic</th>
<th>Uni-directional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Sub-channels</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Optimality

Total number of dimensions required to achieve \( \mathbf{d} \in \mathbb{D} \):

\[
N_s = \begin{cases} 
2 \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{ij}^b + 2 \sum_{j=2}^{3} d_{1j}^c + \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij}^u & \text{bi-directional} \\
3 \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{i=1}^{3} \sum_{j=1}^{3} d_{ij}^c & \text{cyclic} \\
3 \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij}^u & \text{uni-directional}
\end{cases}
\]

\[
= \sum_{i=1}^{3} \sum_{j=1, j \neq i}^{3} d_{ij} - \sum_{i=1}^{3} \sum_{j=i+1}^{3} d_{ij}^b - \sum_{j=2}^{3} d_{1j}^c = d_{ij} + d_{ji} - d_{ij}^b = \max\{d_{ij}, d_{ji}\}
\]

\[
= \max\{d_{12}, d_{21}\} + \max\{d_{13}, d_{31}\} + \max\{d_{23}, d_{32}\} - d_{12}^c - d_{13}^c
\]

\[
d_{12} + d_{23} + d_{31} \text{ e.g. } \Rightarrow d_{13}^c = 0, \quad d_{12}^b = d_{21}
\]

\[
= \max\{d_{12} + d_{23} + d_{31} - d_{12}^c, d_{12} + d_{23} + d_{31} - d_{12}^b\}
\]

\[
= d_{21} + d_{23} + d_{31}
\]

No cycles \( \Rightarrow N_s \leq N \Rightarrow \) All \( \mathbf{d} \in \mathbb{D} \) are achievable
Consider any reliable scheme for the 4-user MIMO MRC
Users can decode their desired signals
Outer bound

Give $m_{23}$ and $y_2$ to user 1 as side info.
Now, user 1 has the info. available at user 2
Outer bound

⇒ User 1 can decode \( m_{32} \)
Upper bound

User 1 can decode \((m_{21}, m_{31}, m_{32})\) from \((m_{12}, m_{13}, y_1, m_{23}, y_2)\)
Upper bound

User 1 can decode \((m_{21}, m_{31}, m_{32})\) from \((m_{12}, m_{13}, y_1, m_{23}, y_2)\).

\[
\Rightarrow R_{21} + R_{31} + R_{32} \leq I(x_r; y_1, y_2) = I\left(x_r; \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} x_r + \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)
\]

P2P Channel
Upper bound

User 1 can decode \((m_{21}, m_{31}, m_{32})\) from \((m_{12}, m_{13}, y_1, m_{23}, y_2)\)

\[\Rightarrow R_{21} + R_{31} + R_{32} \leq I(x_r; y_1, y_2) = I(x_r; \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} x_r + \begin{bmatrix} z_1 \\ z_2 \end{bmatrix})\]

P2P Channel

\[\Rightarrow d_{21} + d_{31} + d_{32} \leq \text{rank} \left( \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \right) = N\]
Upper bound

User 1 can decode \((m_{21}, m_{31}, m_{32})\) from \((m_{12}, m_{13}, y_1, m_{23}, y_2)\) side info.

\[ m_{12}, m_{13}, m_{23} \]

\[ m_{21}, m_{31}, m_{32} \]

\[ \text{Dec1} \]

\[ y_1 \]

\[ D_1 \]

\[ y_2 \]

\[ D_2 \]

\[ \text{Relay} \]

\[ \Rightarrow R_{21} + R_{31} + R_{32} \leq I(x_r; y_1, y_2) = I(x_r; \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} x_r + \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}) \]

P2P Channel

\[ \Rightarrow d_{21} + d_{31} + d_{32} \leq \text{rank} \left( \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \right) = N \]

Considering different combinations of users gives the desired outer bound

\[ \sum_{i=1}^{2} \sum_{j=i+1}^{3} d_{p_i p_j} \leq N, \quad \forall p \]
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \)
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \) ⇒ 2-cycles and 3-cycles!
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow \text{2-cycles and 3-cycles!} \)

**Bi-directional:**
1) set \( d_{ij}^b = d_{ji}^b = \min\{d_{ij}, d_{ji}\} \)
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
Consider a DoF tuple \( d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \) ⇒ 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
Resource allocation

Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**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 \( d_{ij}' = d_{ij} - d_{ij}^b \)
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!
Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**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.) \( \mathbf{d}' = (d_{12}', 0, 0, d_{23}', d_{31}', 0) \Rightarrow 3\text{-cycle}! \)

**Cyclic:**
1) set \( d_{ij}^c = d_{jk}^c = d_{ki}^c = \min\{d_{ij}', d_{jk}', d_{ki}'\} \)
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 \( d_{ij}^b = d_{ji}^b = \min\{d_{ij}, d_{ji}\} \)
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
Resource allocation

Consider a DoF tuple $d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2$-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.) $d' = (d_{12}', 0, 0, d_{23}', d_{31}', 0) \Rightarrow 3$-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
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 $d''_{ij} = d'_{ij} - d_{ij}^c$
Resource allocation

Consider a DoF tuple \( \mathbf{d} = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \Rightarrow 2\text{-cycles and 3\text{-cycles}}! \)

**Bi-directional:**
1) set \( d_{i,j}^b = d_{j,i}^b = \min\{d_{i,j}, d_{j,i}\} \)
2) requires \( d_{i,j}^b \) sub-channels
3) resolves 2-cycles
4) residual DoF \( d_{i,j}' = d_{i,j} - d_{i,j}^b \)

Residual DoF tuple (e.g.) \( \mathbf{d}' = (d_{12}', 0, 0, d_{23}', d_{31}', 0) \Rightarrow 3\text{-cycle}! \)

**Cyclic:**
1) set \( d_{i,j}^c = d_{j,k}^c = d_{k,i}^c = \min\{d_{i,j}', d_{j,k}', d_{k,i}'\} \)
2) requires \( 2d_{i,j}^c \) sub-channels
3) resolves 3-cycles
4) residual DoF \( d_{i,j}'' = d_{i,j} - d_{i,j}^c \)

**Uni-directional:**
1) set \( d_{i,j}^u = d_{i,j}'' \)
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.) \( \mathbf{d}' = (d_{12}', 0, 0, d_{23}', d_{31}' \Rightarrow 3\)-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
Consider a DoF tuple \( d = (d_{12}, d_{13}, d_{21}, d_{23}, d_{31}, d_{32}) \) \( \Rightarrow \) 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.) \( d' = (d'_{12}, 0, 0, d'_{23}, d'_{31}, 0) \) \( \Rightarrow \) 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

\( d \) achieved!