Capacity of $1$-to-$K$ Broadcast Packet Erasure Channels with Channel Output Feedback
— A Packet Evolution Approach

Chih-Chun Wang, Purdue University

Presented in the 48-th Allerton Conference, 9/30/2010

Joint work with Y. Charlie Hu (Purdue), Ness B. Shroff (The OSU), Dimitrios Koutsonikolas, Abdallah Khreishah.

Sponsored by NSF CCF-0845968 and CNS-0905331.
Two Ingredients

Packet Erasure Channels (PECs):

- Input: $X \in \text{GF}(2^b)$ for large $b$.
- A packet $X$ either arrives perfectly (with the help of CRC), or is considered as erasure and discarded. (No hybrid ARQ).
- Memoryless, time-invariant.
Packet Erasure Channels (PECs):

- Input: $X \in \text{GF}(2^b)$ for large $b$.
- A packet $X$ either arrives perfectly (with the help of CRC), or is considered as erasure and discarded. (No hybrid ARQ).
- Memoryless, time-invariant.

The ER protocol — 1-hop cellular networks [Rozner et al. 07].

- 5 transmissions w/o coding vs. 4 transmissions w. coding
- Create its own SI through spatial diversity.
- Empirically, 10–20% throughput improvement.
Two Ingredients

Packet Erasure Channels (PECs):

- Input: \( X \in \text{GF}(2^b) \) for large \( b \).
- A packet \( X \) either arrives perfectly (with the help of CRC), or is considered as erasure and discarded. (No hybrid ARQ).
- Memoryless, time-invariant.

The ER protocol — 1-hop cellular networks [Rozner et al. 07].

- 5 transmissions w/o coding vs. 4 transmissions w. coding
- Create its own SI through spatial diversity.
- Empirically, 10–20% throughput improvement.

Our goal: Finding the Shannon capacity of PECs with channel output feedback (COF) for arbitrary number \( M \geq 3 \) of sessions.

Wang, Allerton 2010 – p. 2/20
The benefits of ER follows from the channel output feedback (COF).

Capacity-achieving schemes by code alignment.

<table>
<thead>
<tr>
<th># of sessions</th>
<th>ER-like Protocols (Broadcast PECs w. COF)</th>
<th>Gaussian broadcast channels w. COF</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=2</td>
<td>Full capacity region [Georgiadis et al. 09]</td>
<td>Outer and inner bounds [Ozarow 84]</td>
</tr>
<tr>
<td>M=3</td>
<td>Full capacity region</td>
<td></td>
</tr>
<tr>
<td>General M</td>
<td>(1) Capacity for fair systems; (2) Outer and inner bounds that meet numerically.</td>
<td></td>
</tr>
</tbody>
</table>
The benefits of ER follows from the channel output feedback (COF).

<table>
<thead>
<tr>
<th># of sessions</th>
<th>ER-like Protocols (Broadcast PECs w. COF)</th>
<th>Gaussian broadcast channels w. COF</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=2</td>
<td>Full capacity region [Georgiadis et al. 09]</td>
<td>Outer and inner bounds [Ozarow 84]</td>
</tr>
<tr>
<td>General M</td>
<td>(1) Capacity for fair systems; (2) Outer and inner bounds that meet numerically.</td>
<td>?</td>
</tr>
</tbody>
</table>

Capacity-achieving schemes by code alignment.

- The problem setting.
- Existing results for $M = 2$ [Georgiadis et al. 09].
- New concepts of code alignment and packet evolution.
- Main theorems and numerical evaluation.
1-Hop Cellular (AP) Networks

- 1-hop access point networks. $M$ dest.
- $M$ can be large, say $\approx 20$.
- Each session has $nR_i$ packets.
- The source $s$ uses the channel $n$ times.

Our goal is to maximize the achievable rate vector $(R_1, \cdots, R_M)$. 

(Instant) feedback

Broadcast PEC

$X_{nR_1}$ $Y_{nR_2}$ $Z_{nR_3}$

$d_1$ $d_2$ $d_3$
Formal Definition of Feasibility

A network code is defined by the following functions:

\[ Y(t) = f_t(\{X_{k,l} : k \in [M], l \in [nR_k]\}, \{Z_k(\tau) : k \in [M], \tau \in [t-1]\}) , \]

\[ \hat{X}_k = g_k(\{Z_k(\tau) : \tau \in [n]\}). \]
A network code is defined by the following functions:
\[ Y(t) = f_t(\{X_{k,l} : k \in [M], l \in [nR_k]\}, \{Z_k(\tau) : k \in [M], \tau \in [t-1]\}), \]
\[ \hat{X}_k = g_k(\{Z_k(\tau) : \tau \in [n]\}). \]

**Definition 1** \((R_1, \ldots, R_M)\) is achievable if \(\forall \epsilon > 0\), there exist a sufficiently large \(n\), a sufficiently large finite field \(\mathbb{F}(2^b)\), and a corresponding network code, such that for independently and uniformly distributed \(X_k, k \in [M]\):
\[
\max_{k \in [M]} P(\hat{X}_k \neq X_k) < \epsilon.
\]
1-Hop Cellular (AP) Networks

- 1-hop access point networks. $M$ dest.
- $M$ can be large, say $\approx 20$.
  (For 2-hop relay networks $M \leq 6$).
- Each session has $nR_i$ packets.
- The source $s$ uses the channel $n$ times.
1-Hop Cellular (AP) Networks

- 1-hop access point networks. $M$ dest.
- $M$ can be large, say $≈ 20$.
- (For 2-hop relay networks $M \leq 6$).
- Each session has $nR_i$ packets.
- The source $s$ uses the channel $n$ times.

For $M = 2$, no feedback, the capacity is $\frac{R_1}{p_1} + \frac{R_2}{p_2} \leq 1$. 

Wang, Allerton 2010 – p. 6/20
1-hop Cellular (AP) Networks

- 1-hop access point networks. \( M \) dest.
- \( M \) can be large, say \( \approx 20 \).
- (For 2-hop relay networks \( M \leq 6 \)).
- Each session has \( nR_i \) packets.
- The source \( s \) uses the channel \( n \) times.
- For \( M = 2 \), no feedback, the capacity is \( \frac{R_1}{p_1} + \frac{R_2}{p_2} \leq 1 \).
- For \( M = 2 \), w. feedback, the capacity is [Georgiadis et al. 09].

\[
\begin{align*}
\frac{R_1}{p_1} + \frac{R_2}{p_2} &\leq 1 \\
\frac{R_1}{p_1} + \frac{R_2}{p_1 \cup 2} &\leq 1
\end{align*}
\]
Outer bound [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

\[
\frac{R_1}{p_1} + \frac{R_2}{p_1 \cup 2} \leq 1 \quad p_2 \to p_{1 \cup 2}
\]

\[
\frac{R_1}{p_{1 \cup 2}} + \frac{R_2}{p_2} \leq 1
\]

The cap. of the original CH with feedback
≺ The cap. of the new physically degraded CH with feedback
= The cap. of the new physically degraded CH without feedback
Georgiadis’ Proof

- **Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

- **Inner bound**: A 2-phase approach. (Creating its own side info.)
Georgiadis’ Proof

**Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

**Inner bound:**

\[
\frac{R_1}{p_1} + \frac{R_2}{p_{1\cup 2}} \leq 1
\]

Phase 1

Rank = \( nR_1 \)

Keep sending until

Phase 2

\[
\frac{R_1}{p_{1\cup 2}} + \frac{R_2}{p_2} \leq 1
\]

either

or

recvd by 1

recvd by 2

Wang, Allerton 2010 – p. 7/20
Georgiadis’ Proof

**Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

**Inner bound:**

\[
\frac{R_1}{p_1} + \frac{R_2}{p_1 \cup p_2} \leq 1
\]

**Phase 1**
- Rank = \( n R_1 \)
- Keep sending until
- Receive either by 1 or 2

**Phase 2**
- Rank = \( n R_2 \)
- Keep sending until
- Receive either by 1 or 2

Wang, Allerton 2010 – p. 7/20
**Georgiadis’ Proof**

- **Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

- **Inner bound:**

  \[
  \frac{R_1}{p_1} + \frac{R_2}{p_{1∪2}} \leq 1 \]

  **Phase 1**
  - Rank = \( nR_1 \)
  - Keep sending until

  **Phase 2**
  - Rank = \( nR_2 \)
  - Either

  - Rcv’d by 1
  - or

  - Rcv’d by 12

  **Mixing**
  - Rcv’d by 2

  - Keep sending until
Georgiadis’ Proof

- **Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

- **Inner bound:**

  $\frac{R_1}{p_1} + \frac{R_2}{p_{1\cup2}} \leq 1$
  $\frac{R_1}{p_{1\cup2}} + \frac{R_2}{p_2} \leq 1$

---

**Phase 1**
- Rank = $nR_1$
- Keep sending until either
- Rank = $nR_2$
- Keep sending until

**Phase 2**
- Rcv'd by 1
- Rcv'd by 2
- Mixing
- Keep sending until both receivers are satisfied

---

Wang, Allerton 2010 – p. 7/20
Georgiadis’ Proof

**Outer bound** [Ozarow et al. 84]: Introduce auxiliary pipes to convert it into *physically degraded channels*, for which feedback does not increase the capacity [El Gamal 78].

**Inner bound:**

\[
\frac{R_1}{p_1} + \frac{R_2}{p_{1 \cup 2}} \leq 1
\]

**Phase 1**
- Rank = \(nR_1\)
- Keep sending until

**Phase 2**
- Rank = \(nR_2\)
- Mix

This scheme achieves the capacity.

Wang, Allerton 2010 – p. 7/20
What if $M \geq 3$?
What if $M \geq 3$?

- The CH. parameters become more involved.
- $M = 2$: $p_{12}, p_{12c}, p_{1c2}, p_{1c2c}$.
- $M \geq 3$: the success probability $p_S([M] \setminus S)$ that a packet is received by and only by $d_i \in S$. We have $2^M$ such parameters.
What if $M \geq 3$?

- The CH. parameters become more involved.
- $M = 2$: $p_{12}$, $p_{12c}$, $p_{1c2}$, $p_{1c2c}$.
- $M \geq 3$: the success probability $p_S([M] \setminus S)$ that a packet is received by and only by $d_i \in S$. We have $2^M$ such parameters.
- Can we also quantify the Shannon capacity for $M \geq 3$?
What if $M \geq 3$?

- The CH. parameters become more involved.
  - $M = 2$: $p_{12}, p_{12c}, p_{1c2}, p_{1c2c}$.
  - $M \geq 3$: the success probability $p_S([M] \setminus S)$ that a packet is received by and only by $d_i \in S$. We have $2^M$ such parameters.

Can we also quantify the Shannon capacity for $M \geq 3$?

- Generalization of the outer bound is straightforward.
- Generalization of the inner bound is more difficult.
For any permutation $\pi : [M] \mapsto [M]$,

- $p_k$: The marginal success probability.
For any permutation $\pi: [M] \mapsto [M]$,

$p_k$: The marginal success probability.
For any permutation $\pi : [M] \mapsto [M],$

- Cap. of the original CH with feedback
- Cap. of the new CH with feedback
- Cap. of the new CH without feedback

$p_k$: The marginal success probability.
Simple Cap. Outer Bound

- Cap. of the original CH with feedback
- Cap. of the new CH with feedback
- Cap. of the new CH without feedback

For any permutation $\pi : [M] \mapsto [M]$,

$\pi(1)$: The marginal success probability.

$p_{\cup S}$: Prob. at least one $d_i \in S$ is successful.

$S^\pi_k = \{ \pi(j) : \forall j = 1, \cdots, k \}$.
Simple Cap. Outer Bound

For any permutation \( \pi : [M] \mapsto [M] \),

- Cap. of the original CH with feedback \( \prec \) Cap. of the new CH with feedback
- Cap. of the new CH without feedback

\( p_k \): The marginal success probability.
\( p_{\cup S} \): Prob. at least one \( d_i \in S \) is successful.

\[ S_{\pi k} = \{ \pi(j) : \forall j = 1, \cdots, k \}. \]

For each \( \pi \), the capacity of the degraded channel is

\[ \sum_{k=1}^{M} R_{\pi(k)} \frac{p_{\cup S_{\pi k}}}{p_{\cup S_{\pi k}}} \leq 1. \]
Simple Cap. Outer Bound

- For any permutation $\pi : [M] \mapsto [M]$,
  - Cap. of the original CH with feedback
  - $\prec$ Cap. of the new CH with feedback
  - $\equiv$ Cap. of the new CH without feedback

- $p_k$: The marginal success probability.
- $p_{\cup S}$: Prob. at least one $d_i \in S$ is successful.

\[ S_{\pi}^k = \{ \pi(j) : \forall j = 1, \cdots, k \} \]

For each $\pi$, the capacity of the degraded channel is

\[ \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}^k}} \leq 1. \]

A capacity outer bound is thus

\[ \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}^k}} \leq 1. \]
Cap. Inner Bound?

How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p_{\pi} \cup S_{\pi}^k} \leq 1 \)
How to achieve the outer bound: $\forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p \cup S_{\pi}^k} \leq 1$

First try was by [Larsson et al. 06], an $M$-phase approach.
Cap. Inner Bound?

How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p \cup S_{k}^{\pi}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.

**Phase 1**
Creating New Coding Opp.

- rcv’d by 1
- rcv’d by 123
- rcv’d by 123

**Phase 2**
Exploiting Coding Opp.

- rcv’d by 123
- rcv’d by 123
- rcv’d by 123

**Phase 3**
Exploiting Coding Opp.

Wang, Allerton 2010 – p. 10/20
How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p \cup S_{\pi}^k} \leq 1 \)

First try was by [Larsson \textit{et al.} 06], an \( M \)-phase approach.
Cap. Inner Bound?

How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p_{\cup S_{\pi}^k}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.
How to achieve the outer bound: $\forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p_{\cup S_{\pi}^k}} \leq 1$

First try was by [Larsson et al. 06], an $M$-phase approach.

Those that have arrived the intended receivers need not be retransmitted!
How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{P_{\cup S_{\pi}^k}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.
How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p \cup S_{\pi}^k} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.
Cap. Inner Bound?

How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p_{\cup S_{\pi}^{k}}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.
Cap. Inner Bound?

How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p \cup S_{\pi k}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.
How to achieve the outer bound: \( \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi}(k)}{p_{\cup S_{\pi}^k}} \leq 1 \)

First try was by [Larsson et al. 06], an \( M \)-phase approach.

Its performance is strictly bounded away from the outer bound.
What Went Wrong?

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.

Phase 3
Exploiting Coding Opp.
What Went Wrong?

Phase 1
Creating New Coding Opp.

Phase 2

Phase 3

d₁ has Y, Z
d₂ has X, Z

By 2 only

By 1 only

By both 1 and 2
What Went Wrong?

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

By 2 only

By 1 only

By both 1 and 2

$X$

$Y$

$Z$

$X + Y$ received by $123$

$d_1$ has $Y, Z$
$d_2$ has $X, Z$
$d_3$ has $X + Y$
What Went Wrong?

Phase 1
Creating New Coding Opp.

Phase 2

Phase 3

By 2 only

By 1 only

By both 1 and 2

$X + Y$ received by $123$

$d_1$ has $Y, Z$
$d_2$ has $X, Z$
$d_3$ has $X + Y$

Discard it => Suboptimal Recoup it for new coding opp.
What Went Wrong?

Phase 1
Creating New Coding Opp.

Phase 2

Phase 3

Discard it => Suboptimal Recoup it for new coding opp.
We need code alignment [W. ISIT10] in order to recoup the overheard coding opportunities during Phases 2 to M. That is, the overheard coding vector $[X + Y]$ has to remain aligned in the subsequent mixing stages. 

$[\alpha (X + Y) + \beta Z]$ serves all three destinations, but $[\alpha X + \beta Y + \gamma Z]$ does not.
New Cap. Inner Bound

- We need code alignment [W. ISIT10] in order to recoup the overheard coding opportunities during Phases 2 to $M$.

- That is, the overheard coding vector $[X + Y]$ has to remain aligned in the subsequent mixing stages.

- $[\alpha(X + Y) + \beta Z]$ serves all three destinations, but $[\alpha X + \beta Y + \gamma Z]$ does not.

- We propose a new Packet Evolution scheme.

- Each information packet (payload) is expanded to (payload, overhearing status, representative coding vector)

  - overhearing status keeps evolving to create more coding opportunities.
  - representative coding vector keeps evolving to ensure code alignment.
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

X

By 2 only

Y

By 1 only

Z

By both 1 and 2

d₁ has Y, Z

d₂ has X, Z
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

X

Y

Z

By 2 only

By 1 only

By both 1 and 2

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

X + Y

By 3 only

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

d_1 has Y, Z
d_2 has X, Z
d_3 has X + Y
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
+ By 3 only
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

$\text{By 1 only}$

$\text{By both 1 and 2}$

$X$

$Y$

$Z$

$X + Y$

$d_1$ has $Y, Z$

$d_2$ has $X, Z$

$d_3$ has $X + Y$
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2

Phase 3

**Phase 1**
- Creating New Coding Opp.

**Phase 2**
- Exploiting Coding Opp.
- Creating New Coding Opp.

**Phase 3**
- Exploiting Coding Opp.
- Creating New Coding Opp.

**By 1 only**
- $Y$

**By 2 only**
- $X$
  - By both 2 and 3

**By both 1 and 2**
- $Z$

**By 3 only**
- $X + Y$
  - $d_1$ has $Y, Z$
  - $d_2$ has $X, Z$
  - $d_3$ has $X + Y$
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2

By 2 only

+ By 3 only

Phase 3

By 1 only

By both 2 and 3

By both 1 and 2

\[ X + Y \]

\[ X \]

\[ Y \]

\[ Z \]

\[ d_1 \text{ has } Y, Z \]
\[ d_2 \text{ has } X, Z \]
\[ d_3 \text{ has } X + Y \]
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

\[ X + Y \] = By both 2 and 3

By 1 only

By both 1 and 2

\[ X + Y \] = By 3 only

\[ d_1 \text{ has } Y, Z \]
\[ d_2 \text{ has } X, Z \]
\[ d_3 \text{ has } X + Y \]
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

\[ X \]

\[ Y \]

\[ Z \]

Phase 2

\[ X + Y \]

By both 2 and 3

Phase 3

\[ X + Y \]

By 3 only

d₁ has \( Y, Z \)
d₂ has \( X, Z \)
d₃ has \( X + Y \)

By 1 only

\[ Z \]

By both 1 and 2
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.
New representative coding vector

Phase 2

Phase 3

$X + Y$ is indeed heard by $d_3$.

By both 2 and 3

By 1 only

By both 1 and 2

$X$ $Y$ $Z$
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

New over hearing status

By both 2 and 3

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

By 1 only

By 3 only

\[ X + Y \]

\[ X + Y \]

\[ X + Y \]

\[ X + Y \] is indeed heard by \( d_3 \).
\[ X + Y \] is actually not heard by \( d_2 \), but is non-interfering.

Wang, Allerton 2010 – p. 13/20
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

\[ X \]

\[ Y \]

\[ Z \]

New overhearing status

\[ X + Y \] = By both 2 and 3

\[ Y \] By 1 only

\[ X + Y \] is indeed heard by \( d_3 \).
\[ X + Y \] is actually not heard by \( d_2 \), but is non-interfering.
\[ X + Y \] is strictly beneficial for \( d_1 \).

\[ Z \] By both 1 and 2

Phase 2

Phase 3

\[ d_1 \text{ has } Y, Z \]
\[ d_2 \text{ has } X, Z \]
\[ d_3 \text{ has } X + Y \]
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2

Phase 3

New representative coding vector

New overhearing status

$X + Y = \text{By both 2 and 3}$

By 1 only + By 3 only = By both 1 and 3

By both 1 and 2

$d_1$ has $Y, Z$
$d_2$ has $X, Z$
$d_3$ has $X + Y$
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

New representative coding vector

By both 1 and 3

By both 1 and 3

By both 1 and 2

\[ X + Y \]

\[ Z \]

\[ d_1 \text{ has } Y, Z \]
\[ d_2 \text{ has } X, Z \]
\[ d_3 \text{ has } X + Y \]
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

\[ X \]

\[ Y \]

\[ Z \]

\[ X + Y \]

New overhearing status

\[ X + Y \] = By both 2 and 3

By both 1 and 2

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

d_1 \text{ has } Y, Z
d_2 \text{ has } X, Z
d_3 \text{ has } X + Y
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

\[ X \]

\[ Y \]

\[ Z \]

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

New overhearing status

\[ X + Y \]

= By both 2 and 3

\[ Z \]

\[ X + Y \]

= By both 1 and 3

\[ X + Y \] is indeed heard by \( d_3 \).

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

\[ d_1 \text{ has } Y, Z \]
\[ d_2 \text{ has } X, Z \]
\[ d_3 \text{ has } X + Y \]
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

New representative coding vector

New overhearing status

By both 1 and 3

$X + Y$ is indeed heard by $d_3$.

$X + Y$ is actually not heard by $d_1$, but is non-interfering.
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

\[ X \]

\[ Y \]

\[ Z \]

New overhearing status

\[ X + Y \] = By both 2 and 3

\[ X + Y \] = By both 1 and 3

By both 1 and 2

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

\[ X + Y \] is indeed heard by \( d_3 \).

\[ X + Y \] is actually not heard by \( d_1 \), but is non-interfering.

\[ X + Y \] is strictly beneficial for \( d_2 \).
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative
coding vector

\[ X \]

\[ Y \]

\[ Z \]

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

New overhearing
status

\[ X + Y \] = By both 2 and 3

\[ X + Y \] = By both 1 and 3

\[ Z \] By both 1 and 2

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

\[ d_1 \] has \( Y, Z \)
\[ d_2 \] has \( X, Z \)
\[ d_3 \] has \( X + Y \)
The Packet Evolution Scheme

Phase 1
Creating New Coding Opp.

New representative coding vector

$X$

$Y$

$Z$

New overhearing status

$X + Y$ = By both 1 and 3

$X + Y$ = By both 1 and 3

Phase 2
Exploiting Coding Opp.
& Creating New Coding Opp.

Phase 3
Exploiting Coding Opp.
& Creating New Coding Opp.

$X + Y + Z$

$d_1$ has $Y, Z$

$d_2$ has $X, Z$

$d_3$ has $X + Y$

Simultaneously serve all three destinations
An Example of $M = 4$

$M = 4$ sessions, each $d_k$ has $X_{k,1}$ to $X_{k,100}$ packets.

Each $X_{k,l}$, $\forall k \in [4], l \in [100]$ has an overheard status $S(X_{k,l}) \subseteq \{1, 2, 3, 4\}$, and a representative coding vector $v(X_{k,l})$ being a 400-dimensional vector.
**An Example of** $M = 4$

- $M = 4$ sessions, each $d_k$ has $X_{k,1}$ to $X_{k,100}$ packets.
- Each $X_{k,l}$, $\forall k \in [4], l \in [100]$ has an overhearing status $S(X_{k,l}) \subseteq \{1, 2, 3, 4\}$, and a representative coding vector $v(X_{k,l})$ being a 400-dimensional vector.
- Use $S(X_{k,l})$ to decide which packets to be coded together.
  
  Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$.
An Example of $M = 4$

- $M = 4$ sessions, each $d_k$ has $X_{k,1}$ to $X_{k,100}$ packets.
- Each $X_{k,l}$, $\forall k \in [4], l \in [100]$ has an overhearing status $S(X_{k,l}) \subseteq \{1, 2, 3, 4\}$, and a representative coding vector $\mathbf{v}(X_{k,l})$ being a 400-dimensional vector.
- Use $S(X_{k,l})$ to decide which packets to be coded together.
  - Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$
- Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $\mathbf{v}(X_{1,l_1})$ to $\mathbf{v}(X_{3,l_3})$.
  - Generate $\mathbf{v}_{tx}$ by $\mathbf{v}_{tx} = c_1 \mathbf{v}(X_{1,l_1}) + c_2 \mathbf{v}(X_{2,l_2}) + c_3 \mathbf{v}(X_{3,l_3})$.
  - Transmit $Y = \mathbf{v}_{tx}(X_{1,1}, \cdots, X_{4,100})^T$. 
An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.

  Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$

- Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $v(X_{1,l_1})$ to $v(X_{3,l_3})$.

  Generate $v_{tx}$ by $v_{tx} = c_1 v(X_{1,l_1}) + c_2 v(X_{2,l_2}) + c_3 v(X_{3,l_3})$.

  Transmit $Y = v_{tx}(X_{1,1}, \ldots, X_{4,100})^T$. 

An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.
  - Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2,3\}$,
    
    $S(X_{2,l_2}) = \{1,3\}$, and $S(X_{3,l_3}) = \{1,2\}$
  - Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $v(X_{1,l_1})$ to $v(X_{3,l_3})$.
    - Generate $v_{tx}$ by $v_{tx} = c_1v(X_{1,l_1}) + c_2v(X_{2,l_2}) + c_3v(X_{3,l_3})$.
    - Transmit $Y = v_{tx}(X_{1,1}, \cdots, X_{4,100})^T$.

- Upon receiving a feedback, say $\{d_3, d_4\}$ receive $Y$:
  - Augment overhearing status $S(x_{k,l})$ and update representative coding vector $v(x_{k,l})$:

    $S(X_{1,l_1}) \leftarrow S(X_{1,l_1}) \cup \{3,4\} = \{2,3,4\}$, \hspace{1cm} $v(X_{1,l_1}) \leftarrow v_{tx}$
    $S(X_{2,l_2}) \leftarrow S(X_{2,l_2}) \cup \{3,4\} = \{1,3,4\}$, \hspace{1cm} $v(X_{2,l_2}) \leftarrow v_{tx}$
    $S(X_{3,l_3}) \leftarrow S(X_{3,l_3}) \cup \{3,4\} = \{1,2,3,4\}$, \hspace{1cm} $v(X_{3,l_3}) \leftarrow v_{tx}$
An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.
  - Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$
  - Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $v(X_{1,l_1})$ to $v(X_{3,l_3})$.
    - Generate $v_{tx}$ by $v_{tx} = c_1 v(X_{1,l_1}) + c_2 v(X_{2,l_2}) + c_3 v(X_{3,l_3})$.
    - Transmit $Y = v_{tx}(X_{1,1}, \ldots, X_{4,100})^T$.
  - Upon receiving a feedback, say “$\{d_3, d_4\}$ receive $Y$”:
    - Augment overhearing status $S(x_{k,l})$ and update representative coding vector $v(x_{k,l})$:
      
      $S(X_{1,l_1}) \leftarrow S(X_{1,l_1}) \cup \{3, 4\} = \{2, 3, 4\}$, 
      $v(X_{1,l_1}) \leftarrow v_{tx}$
      
      $S(X_{2,l_2}) \leftarrow S(X_{2,l_2}) \cup \{3, 4\} = \{1, 3, 4\}$, 
      $v(X_{2,l_2}) \leftarrow v_{tx}$
      
      $S(X_{3,l_3}) \leftarrow S(X_{3,l_3}) \cup \{3, 4\} = \{1, 2, 3, 4\}$, 
      $v(X_{3,l_3}) \leftarrow v_{tx}$
An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.
  - Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2,3\}$, $S(X_{2,l_2}) = \{1,3\}$, and $S(X_{3,l_3}) = \{1,2\}$
  - Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $\mathbf{v}(X_{1,l_1})$ to $\mathbf{v}(X_{3,l_3})$.
    - Generate $\mathbf{v}_{tx}$ by $\mathbf{v}_{tx} = c_1 \mathbf{v}(X_{1,l_1}) + c_2 \mathbf{v}(X_{2,l_2}) + c_3 \mathbf{v}(X_{3,l_3})$.
    - Transmit $Y = \mathbf{v}_{tx}(X_{1,1}, \cdots, X_{4,100})^T$.
  - Upon receiving a feedback, say \(\{d_3, d_4\}\) receive $Y$:
    - Augment overheard status $S(x_{k,l})$ and update representative coding vector $\mathbf{v}(x_{k,l})$:
      
      \[
      \begin{align*}
      S(X_{1,l_1}) &\leftarrow S(X_{1,l_1}) \cup \{3,4\} = \{2,3,4\}, & \mathbf{v}(X_{1,l_1}) &\leftarrow \mathbf{v}_{tx} \\
      S(X_{2,l_2}) &\leftarrow S(X_{2,l_2}) \cup \{3,4\} = \{1,3,4\}, & \mathbf{v}(X_{2,l_2}) &\leftarrow \mathbf{v}_{tx} \\
      S(X_{3,l_3}) &\leftarrow S(X_{3,l_3}) \cup \{3,4\} = \{1,2,3,4\}, & \mathbf{v}(X_{3,l_3}) &\leftarrow \mathbf{v}_{tx}
      \end{align*}
      \]

Create more coding Opp. Create more coding Opp.
An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.
  - Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}, X_{2,l_2},$ and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$

- Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $\mathbf{v}(X_{1,l_1})$ to $\mathbf{v}(X_{3,l_3})$.
  - Generate $\mathbf{v}_{tx}$ by $\mathbf{v}_{tx} = c_1 \mathbf{v}(X_{1,l_1}) + c_2 \mathbf{v}(X_{2,l_2}) + c_3 \mathbf{v}(X_{3,l_3})$.
  - Transmit $Y = \mathbf{v}_{tx}(X_{1,1}, \ldots, X_{4,100})^T$.

- Upon receiving a feedback, say \{d_3, d_4\} receive Y:

  - Augment overhearing status $S(x_{k,l})$ and update representative coding vector $\mathbf{v}(x_{k,l})$:  
    
    $\begin{align*}
    S(X_{1,l_1}) &\leftarrow S(X_{1,l_1}) \cup \{3, 4\} = \{2, 3, 4\}, & \mathbf{v}(X_{1,l_1}) &\leftarrow \mathbf{v}_{tx} \\
    S(X_{2,l_2}) &\leftarrow S(X_{2,l_2}) \cup \{3, 4\} = \{1, 3, 4\}, & \mathbf{v}(X_{2,l_2}) &\leftarrow \mathbf{v}_{tx} \\
    S(X_{3,l_3}) &\leftarrow S(X_{3,l_3}) \cup \{3, 4\} = \{1, 2, 3, 4\}, & \mathbf{v}(X_{3,l_3}) &\leftarrow \mathbf{v}_{tx}
    \end{align*}$

Create more coding Opp.  

Create more coding Opp.  

$X_{3,l_3}$ has arrived $d_3$
An Example of $M = 4$

- Use $S(X_{k,l})$ to decide which packets to be coded together.
- Suppose we plan to encode sessions 1, 2, 3 together. We choose three packets $X_{1,l_1}$, $X_{2,l_2}$, and $X_{3,l_3}$ such that $S(X_{1,l_1}) = \{2, 3\}$, $S(X_{2,l_2}) = \{1, 3\}$, and $S(X_{3,l_3}) = \{1, 2\}$
- Instead of mixing $X_{1,l_1}$ to $X_{3,l_3}$, we mix $v(X_{1,l_1})$ to $v(X_{3,l_3})$.
  - Generate $v_{tx}$ by $v_{tx} = c_1 v(X_{1,l_1}) + c_2 v(X_{2,l_2}) + c_3 v(X_{3,l_3})$.
  - Transmit $Y = v_{tx}(X_{1,1}, \cdots, X_{4,100})^T$. Achieve Code Alignment

Upon receiving a feedback, say \{\(d_3, d_4\)\} receive $Y$:
- Augment over-hearing status $S(x_{k,l})$ and update representative coding vector $v(x_{k,l})$:
  - $S(X_{1,l_1}) \leftarrow S(X_{1,l_1}) \cup \{3, 4\} = \{2, 3, 4\}$
  - $S(X_{2,l_2}) \leftarrow S(X_{2,l_2}) \cup \{3, 4\} = \{1, 3, 4\}$
  - $S(X_{3,l_3}) \leftarrow S(X_{3,l_3}) \cup \{3, 4\} = \{1, 2, 3, 4\}$

Create more coding Opp. $X_{3,l_3}$ has arrived $d_3$

Achieve Code Alignment

$\mathbf{v}(X_{1,l_1}) \leftarrow v_{tx}$
$\mathbf{v}(X_{2,l_2}) \leftarrow v_{tx}$
$\mathbf{v}(X_{3,l_3}) \leftarrow v_{tx}$
Analysis of The PE Schemes

In the packet evolution scheme, each packet evolves independently.
Analysis of The PE Schemes

- In the packet evolution scheme, each packet evolves independently.

- We can quantify the number of slots that a packet has overhearing status $T$. 

\[
\begin{align*}
\text{Rx 3} & \quad \frac{nR_3}{p_3} - \frac{nR_3}{p_{1,3}} - \frac{nR_3}{p_{1,2,3}} + \frac{nR_3}{p_{1,2,3}} \\
\text{Rx 2} & \quad \frac{nR_3}{p_{2,3}} \\
\text{Rx 1} & \quad \frac{nR_3}{p_{1,2,3}}
\end{align*}
\]
Analysis of The PE Schemes

- In the packet evolution scheme, each packet evolves independently.

- We can quantify the number of slots that a packet has overhearing status $T$. The analysis of PE schemes becomes a time-slot packing problem:
Analysis of The PE Schemes

In the packet evolution scheme, each packet evolves independently.

We can quantify the number of slots that a packet has overhearing status $T$. The joint success prob. $p_{S \{1,2,3\}}$ affects the duration of each status, and thus how to pack them.

The analysis of PE schemes becomes a time-slot packing problem:
Capacity Results $M = 3$

Based on the Packet Evolution method, we have:

**Proposition 1** Consider any 1-to-3 broadcast PEC with channel output feedback with arbitrary parameters $p_S(\{1,2,3\}\setminus S)$ for all $S \subseteq \{1,2,3\}$.

The capacity region is indeed $\forall \pi$, $\sum_{k=1}^{M} \frac{R_{\pi(k)}(k)}{p_{\cup S_{\pi k}}} \leq 1$.

6 facets $\Leftrightarrow$ 6 different permutations $\pi$
### Capacity Results $M \geq 4$

**Outer bound:** \[\forall \pi, \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi_k}}} \leq 1.\]

<table>
<thead>
<tr>
<th>Settings with general $M &gt; 3$ values</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td>General ( P_{S[M] \setminus S} )</td>
<td></td>
</tr>
<tr>
<td><strong>Spatially symmetric</strong> broadcast PECs</td>
<td></td>
</tr>
<tr>
<td><strong>Spatially independent</strong> broadcast PECs</td>
<td></td>
</tr>
</tbody>
</table>
**Capacity Results** \( M \geq 4 \)

Outer bound:

\[
\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S^\pi_k}} \leq 1.
\]

<table>
<thead>
<tr>
<th>Settings with general M&gt;3 values</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong> ( PS[M]{\setminus S} )</td>
<td>* A cap. Inner bound by using LP solvers to find the <strong>tightest</strong> time-slot packing</td>
</tr>
<tr>
<td><strong>Spatially symmetric</strong> broadcast PECs</td>
<td>* <strong>Numerically meets</strong> the outer bound for all our experiments</td>
</tr>
<tr>
<td><strong>Spatially independent</strong> broadcast PECs</td>
<td></td>
</tr>
</tbody>
</table>

Wang, Allerton 2010 – p. 17/20
## Capacity Results $M \geq 4$

Outer bound: $\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}^{k}}} \leq 1.$

<table>
<thead>
<tr>
<th>Settings with general $M&gt;3$ values</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong> $p_{S[M]\setminus S}$</td>
<td>* A cap. Inner bound by using LP solvers to find the tightest time-slot packing</td>
</tr>
<tr>
<td><strong>Spatially symmetric</strong> broadcast PECs</td>
<td>* Numerically meets the outer bound for all our experiments</td>
</tr>
<tr>
<td>$p_{S_{1}[M]\setminus S_{1}} = p_{S_{2}[M]\setminus S_{2}}$ if $</td>
<td>S_{1}</td>
</tr>
<tr>
<td><strong>Spatially independent</strong> broadcast PECs</td>
<td></td>
</tr>
</tbody>
</table>
## Capacity Results $M \geq 4$

**Outer bound:** $\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}k}} \leq 1$.

<table>
<thead>
<tr>
<th>Settings with general $M&gt;3$ values</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong> $p_{S[M] \setminus S}$</td>
<td>* A cap. Inner bound by using LP solvers to find the <strong>tightest</strong> time-slot packing * <strong>Numerically meets</strong> the outer bound for all our experiments</td>
</tr>
<tr>
<td><strong>Spatially symmetric</strong> broadcast PECs $p_{S_1[M] \setminus S_1} = p_{S_2[M] \setminus S_2}$ if $</td>
<td>S_1</td>
</tr>
<tr>
<td><strong>Spatially independent</strong> broadcast PECs</td>
<td></td>
</tr>
</tbody>
</table>

Wang, Allerton 2010 – p. 17/20
### Capacity Results $M \geq 4$

#### Outer bound:

$$\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup_{\pi} S_k}} \leq 1.$$  

**Settings with general $M>3$ values**

- **General**
  $$p_{S[M] \setminus S}$$

**Spatially symmetric broadcast PECs**

- $p_{S_1[M] \setminus S_1} = p_{S_2[M] \setminus S_2}$ if $|S_1| = |S_2|$

**Spatially independent broadcast PECs**

- $$p_{S[M] \setminus S} = \prod_{k \in S} p_k \prod_{j \in [M] \setminus S} (1 - p_j)$$

#### Capacity inner bound results

- *A cap. Inner bound by using LP solvers to find the tightest time-slot packing*
- *Numerically meets the outer bound for all our experiments*

**The inner and outer bounds always meet. → Full capacity region.**
### Capacity Results $M \geq 4$

Outer bound:

$$\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_k}^{\pi}} \leq 1.$$  

### Settings with general $M>3$ values

<table>
<thead>
<tr>
<th>General $P_{S[M] \setminus S}$</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td>* A cap. Inner bound by using LP solvers to find the <strong>tightest time-slot packing</strong></td>
<td></td>
</tr>
<tr>
<td>* <strong>Numerically meets</strong> the outer bound for all our experiments</td>
<td></td>
</tr>
</tbody>
</table>

Spatially symmetric broadcast PECs

$$p_{S_1[M] \setminus S_1} = p_{S_2[M] \setminus S_2}$$ if $|S_1| = |S_2|$

<table>
<thead>
<tr>
<th>Spatially independent broadcast PECs $p_{S[M] \setminus S}$</th>
<th>The inner and outer bounds always meet. → Full capacity region.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{S[M] \setminus S} = \prod_{k \in S} p_k \prod_{j \in [M] \setminus S} (1 - p_j)$</td>
<td><strong>The inner and outer bounds meet when</strong> $(R_1, \ldots, R_M)$ <strong>are one-sided fair</strong> (when $R_1 \approx R_2 \approx \cdots \approx R_M$)</td>
</tr>
</tbody>
</table>
Capacity Results $M \geq 4$

Outer bound: $\forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi_k}}} \leq 1.$

<table>
<thead>
<tr>
<th>Settings with general M&gt;3 values</th>
<th>Capacity inner bound results</th>
</tr>
</thead>
<tbody>
<tr>
<td>General $p_{S[M] \setminus S}$</td>
<td>* A cap. Inner bound by using LP solvers to find the tightest time-slot packing</td>
</tr>
<tr>
<td><strong>Spatially symmetric</strong> broadcast PECs $p_{S_1[M] \setminus S_1} = p_{S_2[M] \setminus S_2}$ if $</td>
<td>S_1</td>
</tr>
<tr>
<td><strong>Spatially independent</strong> broadcast PECs $p_{S[M] \setminus S} = \prod_{k \in S} p_k \prod_{j \in [M] \setminus S}$</td>
<td>The inner and outer bounds always meet. → Full capacity region.</td>
</tr>
</tbody>
</table>

The inner and outer bounds meet when $(R_1, \cdots, R_M)$ are one-sided fair $R_1 \approx R_2 \approx \cdots \approx R_M$
Numerical Evaluation

\[ \forall \pi, \quad \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}^k}} \leq 1. \]

Symmetric spatially independent PECs: \( p_1 = p_2 = \cdots = p_M = p \)

Perfectly fair systems: \( R_1 = R_2 = \cdots = R_M \)

Sum rate capacity \( \sum_{k=1}^{M} R_k \) vs. marginal success prob. \( p \).
Numerical Evaluation

\[ \forall \pi, \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_k^{\pi}}} \leq 1. \]

Symmetric spatially independent PECs: \( p_1 = p_2 = \cdots = p_M = p \)
Perfectly fair systems: \( R_1 = R_2 = \cdots = R_M \)

Sum rate capacity \( \sum_{k=1}^{M} R_k \) vs. marginal success prob. \( p \).

Corollary: When \( M \to \infty \), the channel becomes effectively noiseless. [Larsson et al. 06]
Outer bound: $$\forall \pi, \sum_{k=1}^{M} \frac{R_{\pi(k)}}{p_{\cup S_{\pi}k}} \leq 1.$$ Tight for $$M = 3$$

Settings with general $$M > 3$$ values

**General**

$$p_{\mathcal{S}[M] \setminus \mathcal{S}}$$

**Spatially symmetric broadcast PECs**

$$p_{\mathcal{S}_1[M] \setminus \mathcal{S}_1} = p_{\mathcal{S}_2[M] \setminus \mathcal{S}_2} \text{ if } |\mathcal{S}_1| = |\mathcal{S}_2|$$

**Spatially independent broadcast PECs**

$$p_{\mathcal{S}[M] \setminus \mathcal{S}} = \prod_{k \in \mathcal{S}} p_k \prod_{j \in [M] \setminus \mathcal{S}} (1 - p_j)$$

Capacity inner bound results

* A cap. Inner bound by using LP solvers to find the tightest time-slot packing

* **Numerically meets** the outer bound for all our experiments

The inner and outer bounds always meet. → Full capacity region.

The inner and outer bounds meet when $$(R_1, \cdots, R_M)$$ are one-sided fair (when $$R_1 \approx R_2 \approx \cdots \approx R_M$$)
Definition 2  The one-sidedly fair region $\Lambda_{osf}$ contains all rate vectors $(R_1, \cdots, R_M)$ satisfying

$$\forall i, j \text{ satisfying } p_i < p_j, \text{ we have } R_i(1 - p_i) \geq R_j(1 - p_j).$$

Remark 1: A perfectly fair vector $(R, \cdots, R)$ belongs to $\Lambda_{osf}$.

Remark 2: A proportionally fair vector $(p_1 R, \cdots, p_M R)$ belongs to $\Lambda_{osf}$ if $\min\{p_k : \forall k \in [M]\} \geq 0.5$. 