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

Young-Han Kim

University of California, San Diego

Communication Theory Workshop
Dana Point, California
Tuesday, May 12, 2015
Point-to-point codes for interference channels:
A journey toward high performance at low complexity

Young-Han Kim
University of California, San Diego

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

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

Young-Han Kim
University of California, San Diego

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

CONNECTIVITY IS KING

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

Exabytes per Month

61% CAGR 2013-2018

- Mobile File Sharing (2.9%)
- Mobile M2M (5.7%)
- Mobile Audio (10.6%)
- Mobile Web/Data (11.7%)
- Mobile Video (69.1%)

Figures in parentheses refer to traffic share in 2018.
Source: Cisco VNI Mobile, 2014
Where is wireless going?

Billions of Devices

8% CAGR 2013-2018

- Other Portable Devices (0.3%, 0.3%)
- Tablets (1.3%, 5.0%)
- Laptops (2.1%, 2.6%)
- M2M (4.9%, 19.7%)
- Smartphones (24.9%, 38.5%)
- Non-Smartphones (66.4%, 33.9%)

Figures in parentheses refer to device or connections share in 2013, 2018.
Source: Cisco VNI Mobile, 2014
Where is wireless going?

Mobile data per month

<table>
<thead>
<tr>
<th>Year</th>
<th>Exabytes per Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>1.5 EB</td>
</tr>
<tr>
<td>2014</td>
<td>2.6 EB</td>
</tr>
<tr>
<td>2015</td>
<td>4.4 EB</td>
</tr>
<tr>
<td>2016</td>
<td>7.0 EB</td>
</tr>
<tr>
<td>2017</td>
<td>10.8 EB</td>
</tr>
<tr>
<td>2018</td>
<td>15.9 EB</td>
</tr>
</tbody>
</table>

61% CAGR 2013-2018

Source: Cisco VNI Mobile, 2014

Number of devices

<table>
<thead>
<tr>
<th>Year</th>
<th>Billions of Devices</th>
<th>Non-Smart Devices and Connections</th>
<th>Smart Devices and Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>6</td>
<td>79%</td>
<td>21%</td>
</tr>
<tr>
<td>2014</td>
<td>7</td>
<td>74%</td>
<td>26%</td>
</tr>
<tr>
<td>2015</td>
<td>8</td>
<td>68%</td>
<td>32%</td>
</tr>
<tr>
<td>2016</td>
<td>9</td>
<td>61%</td>
<td>39%</td>
</tr>
<tr>
<td>2017</td>
<td>10</td>
<td>54%</td>
<td>46%</td>
</tr>
<tr>
<td>2018</td>
<td>11</td>
<td>46%</td>
<td>54%</td>
</tr>
</tbody>
</table>

8% CAGR 2013-2018

Percentages refer to device or connections share.
Source: Cisco VNI Mobile, 2014
Where is wireless going?

**Mobile data per month**

<table>
<thead>
<tr>
<th>Year</th>
<th>Exabytes per Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>1.5 EB</td>
</tr>
<tr>
<td>2014</td>
<td>2.6 EB</td>
</tr>
<tr>
<td>2015</td>
<td>4.4 EB</td>
</tr>
<tr>
<td>2016</td>
<td>7.0 EB</td>
</tr>
<tr>
<td>2017</td>
<td>10.8 EB</td>
</tr>
<tr>
<td>2018</td>
<td>15.9 EB</td>
</tr>
</tbody>
</table>

**Number of devices**

<table>
<thead>
<tr>
<th>Year</th>
<th>Billions of Devices</th>
<th>Non-Smart Devices</th>
<th>Smart Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>79%</td>
<td>21%</td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>74%</td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>68%</td>
<td>32%</td>
<td></td>
</tr>
<tr>
<td>2016</td>
<td>61%</td>
<td>39%</td>
<td></td>
</tr>
<tr>
<td>2017</td>
<td>54%</td>
<td>46%</td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td>46%</td>
<td>54%</td>
<td></td>
</tr>
</tbody>
</table>

Source: Cisco VNI Mobile, 2014
Where is wireless going?

Mobile data per month

Exabytes per Month

61% CAGR 2013-2018

<table>
<thead>
<tr>
<th>Year</th>
<th>Exabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>1.5 EB</td>
</tr>
<tr>
<td>2014</td>
<td>2.6 EB</td>
</tr>
<tr>
<td>2015</td>
<td>4.4 EB</td>
</tr>
<tr>
<td>2016</td>
<td>7.0 EB</td>
</tr>
<tr>
<td>2017</td>
<td>10.8 EB</td>
</tr>
<tr>
<td>2018</td>
<td>15.9 EB</td>
</tr>
</tbody>
</table>

Source: Cisco VNI Mobile, 2014

Number of devices

Billions of Devices

8% CAGR 2013-2018

<table>
<thead>
<tr>
<th>Year</th>
<th>Non-Smart Devices and Connections</th>
<th>Smart Devices and Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>21%</td>
<td>79%</td>
</tr>
<tr>
<td>2014</td>
<td>26%</td>
<td>74%</td>
</tr>
<tr>
<td>2015</td>
<td>32%</td>
<td>68%</td>
</tr>
<tr>
<td>2016</td>
<td>39%</td>
<td>61%</td>
</tr>
<tr>
<td>2017</td>
<td>46%</td>
<td>54%</td>
</tr>
<tr>
<td>2018</td>
<td>54%</td>
<td>46%</td>
</tr>
</tbody>
</table>

Percentages refer to device or connections share.

Source: Cisco VNI Mobile, 2014
The Internet of Things

Connect the World
Interference, interference, interference
Interference, interference, interference
Interference, interference, interference

Young-Han Kim (UCSD)

P2P codes for interference channels

CTW 2015
Interference in cellular networks

- Desired signal
- Interference
Interference channel

Encoder 1

$X_1^n \xrightarrow{p(y_1, y_2 | x_1, x_2)} Y_1^n$

Encoder 2

$X_2^n \xrightarrow{Y_2^n}$

Decoder 1

$\hat{M}_1$

Decoder 2

$\hat{M}_2$

$M_1$

$M_2$
• Gaussian interference channel
Interference channel

- Gaussian interference channel
- Network with one dominant interferer
Interference channel

- **Capacity region**: Optimal tradeoff between the rates

\[
R_1 = \frac{1}{n} \log |\text{supp}(M_1)|
\]

\[
R_2 = \frac{1}{n} \log |\text{supp}(M_2)|
\]
Interference channel

\[ M_1 \rightarrow \text{Encoder 1} \rightarrow X_1^n \rightarrow p(y_1, y_2 | x_1, x_2) \rightarrow Y_1^n \rightarrow \text{Decoder 1} \rightarrow \hat{M}_1 \]

\[ M_2 \rightarrow \text{Encoder 2} \rightarrow X_2^n \rightarrow Y_2^n \rightarrow \text{Decoder 2} \rightarrow \hat{M}_2 \]

- **Capacity region**: Optimal tradeoff between the rates

\[ R_1 = \frac{1}{n} \log |\text{supp}(M_1)| \]
\[ R_2 = \frac{1}{n} \log |\text{supp}(M_2)| \]
Interference channel

- **Capacity region**: Optimal tradeoff between the rates

\[
R_1 = \frac{1}{n} \log |\text{supp}(M_1)|
\]

\[
R_2 = \frac{1}{n} \log |\text{supp}(M_2)|
\]
**Capacity region**: Optimal tradeoff between the rates

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

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

Performance benchmark

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

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

- Random code ensemble
  - For each $m_1 \in [1 : 2^{nR_1}]$, generate $X_1^n(m_1) \sim \prod_{i=1}^n p_{X_1}(x_{1i})$
  - For each $m_2 \in [1 : 2^{nR_2}]$, generate $X_2^n(m_2) \sim \prod_{i=1}^n p_{X_2}(x_{2i})$
Performance benchmark

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

- Random code ensemble
  - For each $m_1 \in [1 : 2^{nR_1}]$, generate $X_1^n(m_1) \sim \prod_{i=1}^n p_{X_1}(x_{1i})$
  - For each $m_2 \in [1 : 2^{nR_2}]$, generate $X_2^n(m_2) \sim \prod_{i=1}^n p_{X_2}(x_{2i})$

- What is the optimal (MLD) tradeoff between achievable $R_1$ and $R_2$?
Why do we care?

- **Optimal** when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
Why do we care?

- **Optimal** when interference is weak or strong (Sato 1978, ..., Liu–Nair–Xia 2014)
- **P2P random codes** ≈ COTS (commercial off-the-shelf) codes
Why do we care?

- **Optimal** when interference is *weak* or *strong* (Sato 1978, …, Liu–Nair–Xia 2014)
- **P2P random codes** $\approx$ **COTS** (commercial off-the-shelf) codes
- **Han–Kobayashi** coding scheme
Why do we care?

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

\[
p(y_1, y_2 | x_1, x_2)\]

\[
p(y_1, y_2 | u_1, u_2, v_1, v_2)\]
Why do we care?

- **Optimal** when interference is weak or strong (Sato 1978, …, Liu–Nair–Xia 2014)
- P2P random codes \(\approx\) COTS (commercial off-the-shelf) codes
- Han–Kobayashi coding scheme
Performance benchmark (Bandemer–El-Gamal–K 2012)

\[ p(y_1|x_1, x_2) \]

\[ p(y_2|x_1, x_2) \]
Performance benchmark (Bandemer–El-Gamal–K 2012)

\[ R_1 < I(X_1; Y_1 | X_2), \]
\[ R_1 + R_2 < I(X_1, X_2; Y_1) \]

or

\[ R_1 < I(X_1; Y_1) \]
Performance benchmark (Bandemer–El-Gamal–K 2012)

\[ R_1 < I(X_1; Y_1 | X_2), \]
\[ R_1 + R_2 < I(X_1, X_2; Y_1) \]

or

\[ R_1 < I(X_1; Y_1) \]
Performance benchmark (Bandemer–El-Gamal–K 2012)

\[ R_1 < I(X_1; Y_1 | X_2), \]
\[ R_1 + R_2 < I(X_1, X_2; Y_1) \]

or

\[ R_1 < I(X_1; Y_1) \]
Maximum likelihood decoding $\approx$ simultaneous decoding

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

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

or

$R_1 < I(X_1; Y_1)$
Maximum likelihood decoding \(\approx\) simultaneous decoding

\[
R_1 < I(X_1; Y_1 | X_2),
\]
\[
R_1 + R_2 < I(X_1, X_2; Y_1)
\]

or

\[
R_1 < I(X_1; Y_1)
\]

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

\[ R_1 < I(X_1; Y_1 | X_2), \]
\[ R_1 + R_2 < I(X_1, X_2; Y_1) \]

or

\[ R_1 < I(X_1; Y_1) \]

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

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

\begin{align*}
R_1 &< I(X_1; Y_1 | X_2), \\
R_1 + R_2 &< I(X_1, X_2; Y_1) \\
\text{or} \\
R_1 &< I(X_1; Y_1)
\end{align*}

- Maximum likelihood decoding: $\arg\max \sum_{\hat{m}_1} \sum_{m_2} p(y^n_1 | \hat{m}_1, m_2)$
- Simultaneous ML decoding: $\arg\max \max_{\hat{m}_1} \sum_{m_2} p(y^n_1 | \hat{m}_1, m_2)$
- Simultaneous nonunique decoding (SND): $(x^n_1(\hat{m}_1), x^n_2(m_2), y^n_1) \in \mathcal{T}_e^{(n)}$ for some $m_2$
Maximum likelihood decoding $\approx$ simultaneous decoding

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

or

$R_1 < I(X_1; Y_1)$

- **Maximum likelihood decoding**: $\arg \max \hat{m}_1 \sum_{m_2} p(y_1^n | \hat{m}_1, m_2)$
- **Simultaneous ML decoding**: $\arg \max \hat{m}_1 \max_{m_2} p(y_1^n | \hat{m}_1, m_2)$
- **Simultaneous nonunique decoding (SND)**: $(x_1^n(\hat{m}_1), x_2^n(m_2), y_1^n) \in T_e^{(n)}$ for some $m_2$
- **Multiuser detection**: High complexity!
Low-complexity (implementable) alternatives

\[ p(y_1|x_1, x_2) \]

\[ p(y_2|x_1, x_2) \]
Low-complexity (implementable) alternatives

- P2P decoding
Low-complexity (implementable) alternatives

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

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

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

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

- + rate splitting (Zhao et al. 2011, Wang et al. 2014)

- **Novel codes**
  - **Spatially coupled codes** (Yedla, Nguyen, Pfister, and Narayanan 2011)
  - **Polar codes** (Wang and Şaşoğlu 2014)
Low-complexity (implementable) alternatives

- P2P decoding
  - Treating interference as (Gaussian) noise: \( R_1 < I(X_1; Y_1) \)
  - Successive cancellation decoding: \( R_2 < I(X_2; Y_1) \), \( R_1 < I(X_1; Y_1|X_2) \)
- + rate splitting (Zhao et al. 2011, Wang et al. 2014)
- Novel codes
  - Spatially coupled codes (Yedla, Nguyen, Pfister, and Narayanan 2011)
  - Polar codes (Wang and Şaşoğlu 2014)
A lesson from rate splitting (Grant et al. 2001)
A lesson from rate splitting (Grant et al. 2001)

Decoding at receiver 1:

\[ R'_1 < I(U; Y_1) \]
\[ R_2 < I(X_2; Y_1|U) \]
\[ R''_1 < I(V; Y_1|U, X_2) \]
A lesson from rate splitting (Grant et al. 2001)

- Decoding at receiver 1:

\[ R'_1 < I(U; Y_1) \]
\[ R_2 < I(X_2; Y_1 | U) \]
\[ R''_1 < I(V; Y_1 | U, X_2) \]
A lesson from rate splitting (Grant et al. 2001)

- Decoding at receiver 1:

\[ R_1' < I(U; Y_1) \]
\[ R_2 < I(X_2; Y_1 | U) \]
\[ R_1'' < I(V; Y_1 | U, X_2) \]
A lesson from rate splitting (Grant et al. 2001)

Decoding at receiver 1:

\[ R_1' < I(U; Y_1) \]
\[ R_2 < I(X_2; Y_1 | U) \]
\[ R_1'' < I(V; Y_1 | U, X_2) \]

Decoding at receiver 2:

\[ R_1' < I(U; Y_2) \]
\[ R_2 < I(X_2; Y_2 | U) \]
\[ R_1'' < I(V; Y_2 | U, X_2) \]
A lesson from rate splitting (Grant et al. 2001)

- Decoding at receiver 1:
  \[ R'_1 < I(U; Y_1) \]
  \[ R_2 < I(X_2; Y_1 | U) \]
  \[ R''_1 < I(V; Y_1 | U, X_2) \]

- Decoding at receiver 2:
  \[ R'_1 < I(U; Y_2) \]
  \[ R_2 < I(X_2; Y_2 | U) \]
  \[ R''_1 < I(V; Y_2 | U, X_2) \]
A lesson from rate splitting (Grant et al. 2001)

- **Decoding at receiver 1:** \( R'_1 < I(U; Y_1) \), \( R_2 < I(X_2; Y_1|U) \), \( R''_1 < I(V; Y_1|U, X_2) \)

- **Decoding at receiver 2:** \( R'_1 < I(U; Y_2) \), \( R_2 < I(X_2; Y_2|U) \), \( R''_1 < I(V; Y_2|U, X_2) \)

- **Combined rate:**

\[
R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2)
\]
A lesson from rate splitting (Grant et al. 2001)

- Decoding at receiver 1: \( R'_1 < I(U; Y_1), R_2 < I(X_2; Y_1|U), R''_1 < I(V; Y_1|U, X_2) \)
- Decoding at receiver 2: \( R'_1 < I(U; Y_2), R_2 < I(X_2; Y_2|U), R''_1 < I(V; Y_2|U, X_2) \)
- Combined rate:

\[
R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2) < \min_j [I(U; Y_j) + I(V; Y_j|U, X_2)]
\]
A lesson from rate splitting (Grant et al. 2001)

- Decoding at receiver 1: \( R'_1 < I(U; Y_1), \ R_2 < I(X_2; Y_1|U), \ R''_1 < I(V; Y_1|U, X_2) \)
- Decoding at receiver 2: \( R'_1 < I(U; Y_2), \ R_2 < I(X_2; Y_2|U), \ R''_1 < I(V; Y_2|U, X_2) \)
- Combined rate:
  \[
  R_1 < \min_j I(U; Y_j) + \min_j I(V; Y_j|U, X_2)
  < \min_j [I(U; Y_j) + I(V; Y_j|U, X_2)]
  \]
- Key to achieving the SD performance: Switch the order of sum and min!
Sliding-window superposition coding (Wang et al. 2014)

\[ M_1(j-1) \rightarrow U^n \rightarrow X_1^n \rightarrow p(y_1|x_1, x_2) \rightarrow Y_1^n \rightarrow M_2(j) \rightarrow M_1(j) \]

\[ M_1(j) \rightarrow V^n \rightarrow X_1^n \rightarrow p(y_1|x_1, x_2) \rightarrow Y_1^n \rightarrow M_2(j) \rightarrow M_1(j) \]

\[ M_2(j) \rightarrow X_2^n \rightarrow p(y_2|x_1, x_2) \rightarrow Y_2^n \rightarrow M_2(j) \rightarrow M_1(j) \]
Sliding-window superposition coding (Wang et al. 2014)

\[ p(y_1| x_1, x_2) \]

Block

\[
\begin{array}{cccccccc}
| & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
U & & & & & & & \\
V & & & & & & & \\
X_2 & M_2(1) & M_2(2) & M_2(3) & M_2(4) & M_2(5) & M_2(6) & M_2(7) \\
\end{array}
\]
Sliding-window superposition coding (Wang et al. 2014)

\[ p(y_1|x_1, x_2) \]

\[ p(y_2|x_1, x_2) \]

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( M_1(1) )</td>
</tr>
<tr>
<td>( V )</td>
<td>( M_1(1) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( X_2 )</td>
<td>( M_2(1) )</td>
<td>( M_2(2) )</td>
<td>( M_2(3) )</td>
<td>( M_2(4) )</td>
<td>( M_2(5) )</td>
<td>( M_2(6) )</td>
<td>( M_2(7) )</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

\[ \begin{align*}
M_1(j - 1) & \xrightarrow{U^n} X_1^n \\
M_1(j) & \xrightarrow{V^n} X_1^n \\
M_2(j) & \xrightarrow{X_2^n} \\
\end{align*} \]

\[ p(y_1|x_1, x_2) \]

\[ p(y_2|x_1, x_2) \]

Block 1 2 3 4 5 6 7
\hline
U \[
\begin{array}{cc}
M_1(1) & M_1(2) \\
\end{array}
\]

V \[
\begin{array}{cc}
M_1(1) & M_1(2) \\
\end{array}
\]

X_2 \[
\begin{array}{cccccc}
M_2(1) & M_2(2) & M_2(3) & M_2(4) & M_2(5) & M_2(6) & M_2(7) \\
\end{array}
\]
Sliding-window superposition coding (Wang et al. 2014)

\[
M_1(j-1) \rightarrow U^n \rightarrow X_1^n \rightarrow p(y_1|x_1, x_2) \rightarrow Y_1^n \rightarrow M_2(j) \rightarrow M_1(j)
\]

\[
M_1(j) \rightarrow V^n \rightarrow X_1^n \rightarrow p(y_1|x_1, x_2) \rightarrow Y_2^n \rightarrow M_2(j) \rightarrow M_1(j)
\]

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>M_1(1)</td>
<td>M_1(2)</td>
<td>M_1(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>M_1(1)</td>
<td>M_1(2)</td>
<td>M_1(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_2</td>
<td>M_2(1)</td>
<td>M_2(2)</td>
<td>M_2(3)</td>
<td>M_2(4)</td>
<td>M_2(5)</td>
<td>M_2(6)</td>
<td>M_2(7)</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

Block 1 2 3 4 5 6 7

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_2$</td>
<td>$M_2(1)$</td>
<td>$M_2(2)$</td>
<td>$M_2(3)$</td>
<td>$M_2(4)$</td>
<td>$M_2(5)$</td>
<td>$M_2(6)$</td>
<td>$M_2(7)$</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

![Diagram showing the sliding-window superposition coding process]

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M_1(1) )</td>
<td>( M_1(2) )</td>
<td>( M_1(3) )</td>
<td>( M_1(4) )</td>
<td>( M_1(5) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>( M_1(1) )</td>
<td>( M_1(2) )</td>
<td>( M_1(3) )</td>
<td>( M_1(4) )</td>
<td>( M_1(5) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( X_2 )</td>
<td>( M_2(1) )</td>
<td>( M_2(2) )</td>
<td>( M_2(3) )</td>
<td>( M_2(4) )</td>
<td>( M_2(5) )</td>
<td>( M_2(6) )</td>
<td>( M_2(7) )</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

\[
M_1(j - 1) \quad U^n \quad X^n_1 \\
M_1(j) \quad V^n
\]

\[
M_2(j) \quad X^n_2
\]

\[
p(y_1|x_1, x_2) \quad Y^n_1 \quad M_2(j) \rightarrow M_1(j)
\]

\[
p(y_2|x_1, x_2) \quad Y^n_2 \quad M_2(j) \rightarrow M_1(j)
\]

Block

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccccc}
M_1(1) & M_1(2) & M_1(3) & M_1(4) & M_1(5) & M_1(6) \\
M_1(1) & M_1(2) & M_1(3) & M_1(4) & M_1(5) & M_1(6) \\
M_2(1) & M_2(2) & M_2(3) & M_2(4) & M_2(5) & M_2(6) & M_2(7)
\end{array}
\]
Sliding-window superposition coding (Wang et al. 2014)

Block Markov coding: As in relaying and feedback communication
Sliding-window superposition coding (Wang et al. 2014)

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

Block Markov coding: As in relaying and feedback communication

Superposition coding: But without rate splitting
Sliding-window superposition coding (Wang et al. 2014)

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

**Diagram**:

- Blocks 1 to 7
- Messages $M_1(j-1)$, $M_1(j)$, $M_2(j)$
- Signals $U^n$, $V^n$, $X_1^n$, $X_2^n$
- Outputs $Y_1^n$, $Y_2^n$
- Probability distributions $p(y_1|x_1,x_2)$, $p(y_2|x_1,x_2)$

**Table**: Block assignments

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td>$M_1(5)$</td>
<td>$M_1(6)$</td>
<td></td>
</tr>
<tr>
<td>$V$</td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td>$M_1(5)$</td>
<td>$M_1(6)$</td>
<td></td>
</tr>
<tr>
<td>$X_2$</td>
<td>$M_2(1)$</td>
<td>$M_2(2)$</td>
<td>$M_2(3)$</td>
<td>$M_2(4)$</td>
<td>$M_2(5)$</td>
<td>$M_2(6)$</td>
<td>$M_2(7)$</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

\[ p(y_1| x_1, x_2) \]

\[ p(y_2| x_1, x_2) \]

- Sliding-window decoding

<table>
<thead>
<tr>
<th>Block</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U )</td>
<td>( M_1(1) )</td>
<td>( M_1(2) )</td>
<td>( M_1(3) )</td>
<td>( M_1(4) )</td>
<td>( M_1(5) )</td>
<td>( M_1(6) )</td>
<td></td>
</tr>
<tr>
<td>( V )</td>
<td>( M_1(1) )</td>
<td>( M_1(2) )</td>
<td>( M_1(3) )</td>
<td>( M_1(4) )</td>
<td>( M_1(5) )</td>
<td>( M_1(6) )</td>
<td></td>
</tr>
<tr>
<td>( X_2 )</td>
<td>( M_2(1) )</td>
<td>( M_2(2) )</td>
<td>( M_2(3) )</td>
<td>( M_2(4) )</td>
<td>( M_2(5) )</td>
<td>( M_2(6) )</td>
<td>( M_2(7) )</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

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

- Sliding-window decoding
- Successive cancellation decoding

\[ R_2 < I(X_2; Y_j | U), \]
Sliding-window superposition coding (Wang et al. 2014)

- Sliding-window decoding
- Successive cancellation decoding

\[
R_2 < I(X_2; Y_j|U),
\]

\[
R_1 < I(U; Y_j) + I(V; Y_j|U, X_2)
\]
Sliding-window superposition coding (Wang et al. 2014)

- Every corner point: different decoding orders
Sliding-window superposition coding (Wang et al. 2014)

• Every corner point: **different decoding orders**

• Every point: time sharing or **more superposition layers**

\[
M_1(j-1) \xrightarrow{U^n} X_1^n \xrightarrow{p(y_1|x_1,x_2)} Y_1^n \xrightarrow{M_2(j) \rightarrow M_1(j)}
\]

\[
M_1(j) \xrightarrow{V^n} X_1^n \xrightarrow{p(y_1|x_1,x_2)} Y_1^n \xrightarrow{M_2(j) \rightarrow M_1(j)}
\]

\[
M_2(j) \xrightarrow{X_2^n} \xrightarrow{p(y_2|x_1,x_2)} Y_2^n \xrightarrow{M_2(j) \rightarrow M_1(j)}
\]

Block

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
<td></td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td>$M_1(5)$</td>
<td>$M_1(6)$</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td>$M_1(1)$</td>
<td>$M_1(2)$</td>
<td>$M_1(3)$</td>
<td>$M_1(4)$</td>
<td>$M_1(5)$</td>
<td>$M_1(6)$</td>
<td>$M_1(6)$</td>
</tr>
<tr>
<td>$X_2$</td>
<td>$M_2(1)$</td>
<td>$M_2(2)$</td>
<td>$M_2(3)$</td>
<td>$M_2(4)$</td>
<td>$M_2(5)$</td>
<td>$M_2(6)$</td>
<td>$M_2(7)$</td>
</tr>
</tbody>
</table>
Sliding-window superposition coding (Wang et al. 2014)

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

\[ U \quad M' \quad V \quad M'' \]

**Multilevel coding (MLC)**

\[ R' < I(U; Y) \]
\[ R'' < I(V; Y|U) \]

Short, nonuniversal
Towards coded modulation

\[ U \quad X \quad V \]

- Multilevel coding (MLC)
- Bit-interleaved coded modulation (BICM)

\[ R' < I(U; Y) \]
\[ R'' < I(V; Y|U) \]

\[ R < I(U; Y) + I(V; Y) \]

Short, nonuniversal
Other layers as noise
Towards coded modulation

Multilevel coding (MLC)

\[ R' < I(U; Y) \]
\[ R'' < I(V; Y | U) \]

Short, nonuniversal

Bit-interleaved coded modulation (BICM)

\[ R < I(U; Y) + I(V; Y) \]

Other layers as noise

Sliding-window coded modulation (SWCM)

\[ R < I(U; Y) + I(V; Y | U) = I(X; Y) \]

Error prop., rate loss

Young-Han Kim (UCSD)
P2P performance

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

- Every corner point: different decoding orders
- Every point: time sharing or more superposition layers
- General theory for arbitrary number of users (Wang 2015)
Gaussian channel performance (Park–K–Wang 2014)

LTE turbo code / \leq 8\text{-iteration} LOG-MAP decoding at $b = 20, n = 2048, \text{BLER} = 0.1, \text{SNR} = 10 \text{ dB}$
LTE turbo code / ≤8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER $= 0.1$
LTE turbo code / $\leq$8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER = 0.1
LTE turbo code / ≤ 8-iteration LOG-MAP decoding at $b = 20$, $n = 660$ (13200 REs), BLER = 0.1
Cooper’s Law

Source: Arraycomm, Zander–Mähonen (2013)
Cooper’s Law

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

Source: Arraycomm, Zander–Mähönen (2013)
Cooper’s Law

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

Source: Arraycomm, Zander–Mähönen (2013)
Cooper’s Law

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

- Spectral efficiency $\eta$: $x 25$
- System bandwidth $W_{sys}$: $x 25$

Source: Arraycomm, Zander–Mähönen (2013)
Cooper’s Law

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

Source: Arraycomm, Zander–Mähönen (2013)
What’s next?

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

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

- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding
What’s next?

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

- Network coding
- Noisy network coding
- Distributed decode–forward
- Index coding
- Distributed storage coding
What’s next?

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

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

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

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

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

Let’s have fun building better networks!