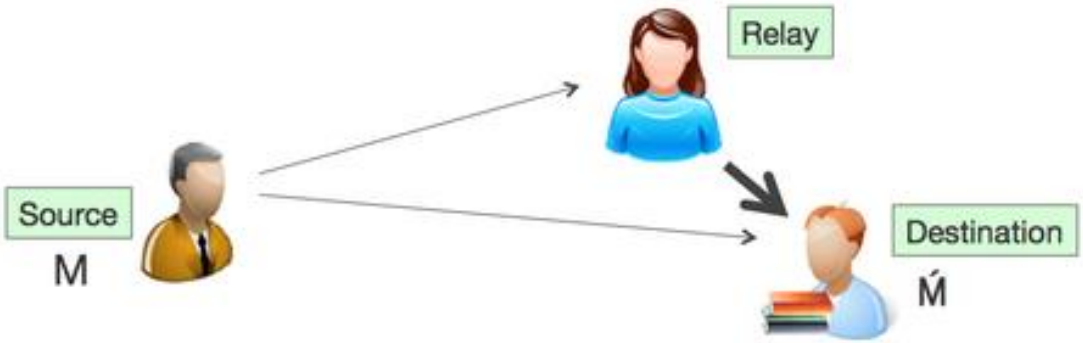
Precoder Detection for Cooperative Decode-and-Forward Relaying in OFDMA Systems

Abhijit Kiran Valluri, Richard J. La, Mark A. Shayman

UNIVERSITY OF MARYLAND
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

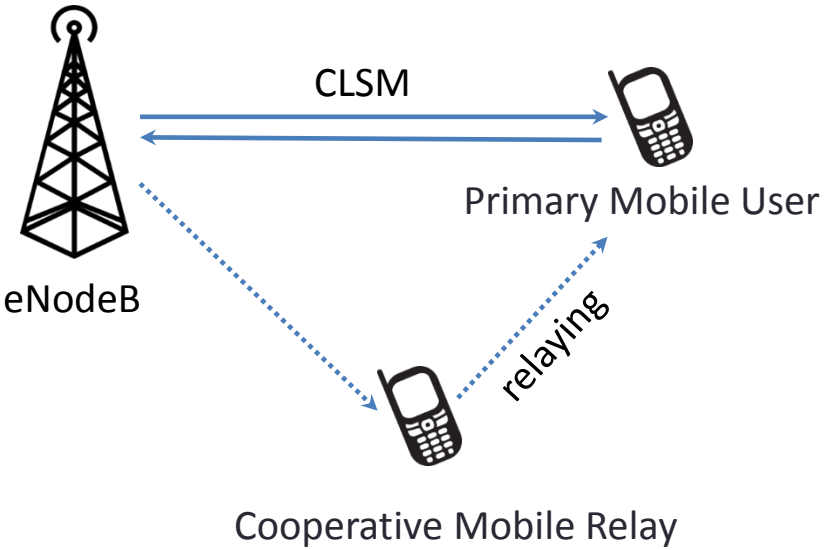
Problem Description

A cooperative relay scenario in LTE



Problem Description

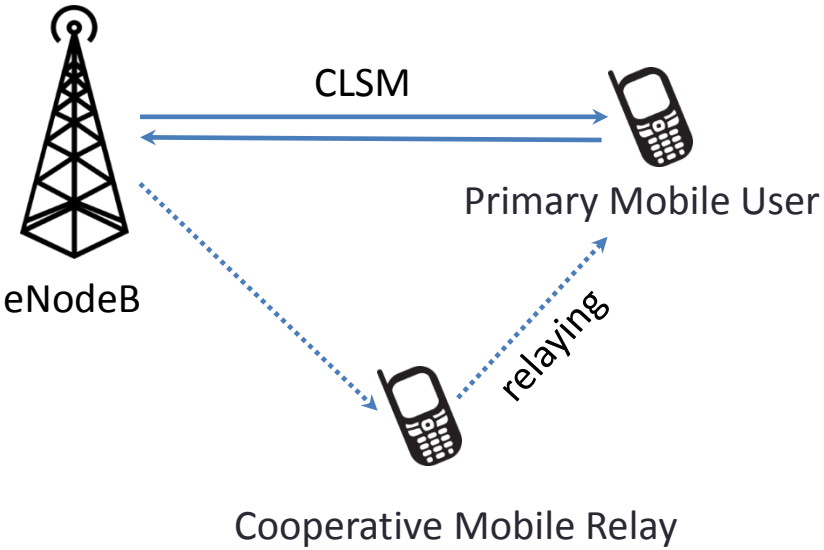
A cooperative relay scenario in LTE



- Secondary user acts as a Cooperative Mobile Relay (CMR)
- Uses decode-and-forward relaying mechanism with Primary User (PU)
- CMR needs to detect **unknown** Precoder

Problem Description

A cooperative relay scenario in LTE

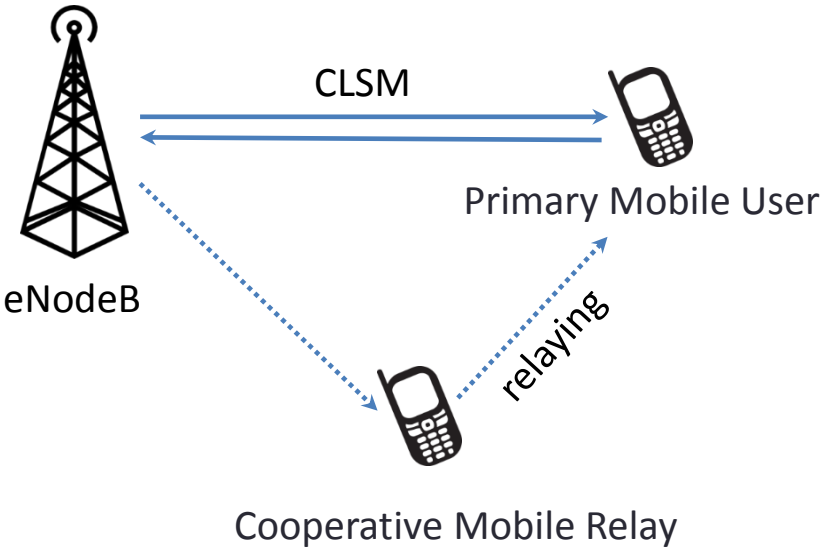


- Secondary user acts as a Cooperative Mobile Relay (CMR)
- Uses decode-and-forward relaying mechanism with Primary User (PU)
- CMR needs to detect **unknown** Precoder

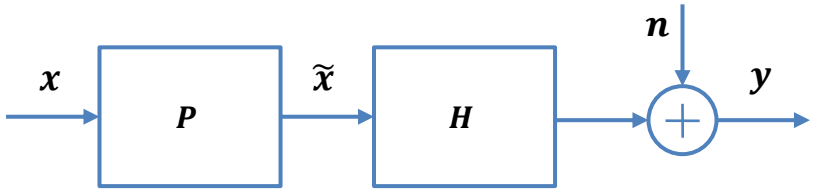
$$\left\{ \boldsymbol{x} \rightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \rightarrow \tilde{\boldsymbol{x}} \right\}$$

Problem Description

A cooperative relay scenario in LTE



- Secondary user acts as a Cooperative Mobile Relay (CMR)
- Uses decode-and-forward relaying mechanism with Primary User (PU)
- CMR needs to detect **unknown** Precoder



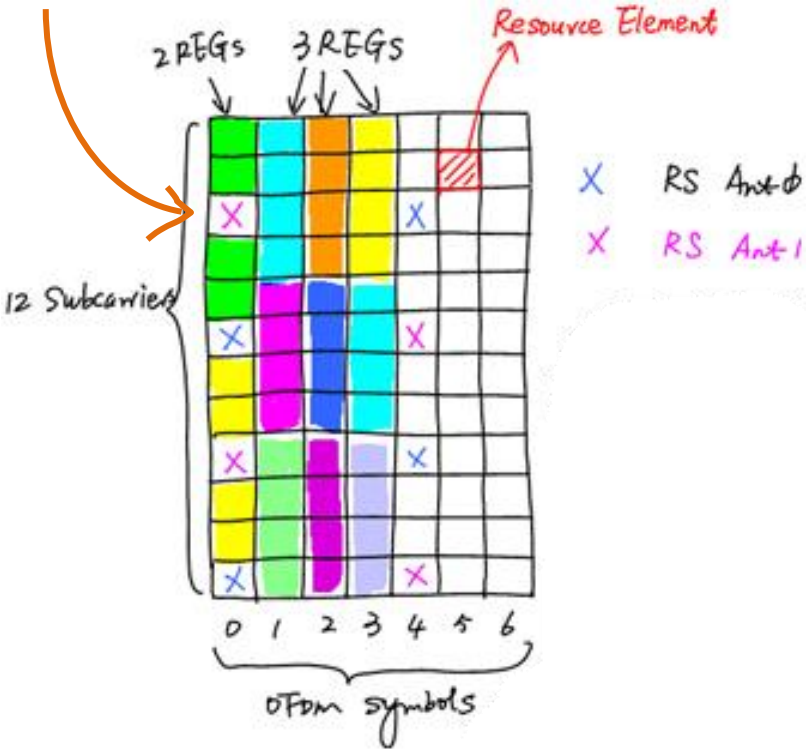
A simple transmission model

Motivation

- In Closed-Loop Spatial Multiplexing (CLSM), downlink transmission uses antenna ports {0, 1, 2, 3}
  - Use Cell-specific Reference Signals (CRSs)
  - CRSs added **after** precoding

Motivation

- In Closed-Loop Spatial Multiplexing (CLSM), downlink transmission uses antenna ports {0, 1, 2, 3}
  - Use Cell-specific Reference Signals (CRSs)
  - CRSs added **after** precoding



Motivation

- In Closed-Loop Spatial Multiplexing (CLSM), downlink transmission uses antenna ports {0, 1, 2, 3}
  - Use Cell-specific Reference Signals (CRSs)
  - CRSs added **after** precoding
- PU uses CRSs and knowledge of precoder used to demodulate received signal
- Precoder information is not always specified by eNodeB
  - PU knows the precoder as it chose it
  - CMR is not aware of the precoder

Precoder Detection

- **Goal:** Design an algorithm to detect the employed precoder in downlink transmission from eNodeB

- **Approaches**
  - Hypothesis testing Framework
    - Simplified ML algorithm
    - Cluster variance algorithm

Precoder Detection

Received OFDM data symbols:

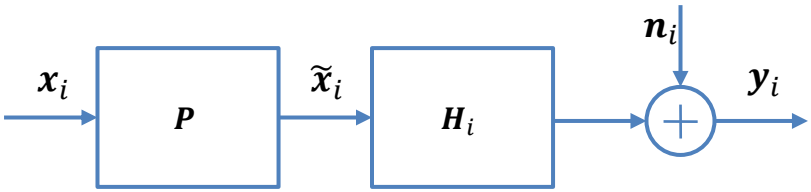
$$\boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{P} \boldsymbol{x}_i + \boldsymbol{n}_i$$

$\boldsymbol{x}_i$ – i^{th} transmitted OFDM data symbol

$\boldsymbol{H}_i$ – frequency domain channel gain matrix for $\boldsymbol{x}_i$

$\boldsymbol{P}$ – a precoder matrix from set $\{\boldsymbol{P}_j, j \in \{1, \dots, N\}\}$

$\boldsymbol{n}_i$ – noise (i.i.d., zero mean Gaussian)



Precoder Detection

Hypothesis testing framework:

$$\left\{ \boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{P} \boldsymbol{x}_i + \boldsymbol{n}_i \right\}$$

- A naïve Maximum Likelihood (ML) hypothesis test:

$$\mathcal{L}(\boldsymbol{P}_j | \boldsymbol{H}_i, \boldsymbol{y}_i) = \underbrace{\frac{\mathbf{1}}{|\chi|} \sum_{\boldsymbol{x}_i \in \chi} f(\boldsymbol{y}_i \mid \boldsymbol{H}_i, \boldsymbol{P}_j, \boldsymbol{x}_i)}$$

$$\mathcal{L}(\boldsymbol{P}_j | \boldsymbol{H}, \boldsymbol{y}) = \prod_{i=1}^{M} \mathcal{L}(\boldsymbol{P}_j | \boldsymbol{H}_i, \boldsymbol{y}_i)$$

χ – set of all possible transmit vectors; can be very large!

$\boldsymbol{H} := (\boldsymbol{H}_1, \dots, \boldsymbol{H}_M), \boldsymbol{y} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_M)$

Precoder Detection – Simplified ML

$$\left\{ \boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{P} \boldsymbol{x}_i + \boldsymbol{n}_i \right\}$$

- Equalize $\boldsymbol{y}_i$ using an MMSE filter to get $\widetilde{\boldsymbol{y}}_i$
- Decode $\boldsymbol{x}_i$ as $\widehat{\boldsymbol{x}}_i^{(j)}$ assuming $\boldsymbol{P}_j$ is true precoder matrix:

$$\widehat{\boldsymbol{x}}_i^{(j)} := \underset{\boldsymbol{x} \in \chi}{\operatorname{argmin}} \left\| \boldsymbol{P}_j^{+} \widetilde{\boldsymbol{y}}_i - \boldsymbol{x} \right\|$$

- Derive a simplified log-likelihood function:

$$\log \mathcal{L}\left(\boldsymbol{P}_j \middle| \widehat{\boldsymbol{H}}, \boldsymbol{y}, \widehat{\boldsymbol{x}}^{(j)} \right) = -\sum_{i=1}^{M} \frac{1}{2\widehat{\sigma}^2} \left\| \boldsymbol{y}_i - \widehat{\boldsymbol{H}}_i \boldsymbol{P}_j \widehat{\boldsymbol{x}}_i^{(j)} \right\|^2 + \text{const.}$$

- Choose $\boldsymbol{P}_j$ that maximizes above log-likelihood function

Precoder Detection – Simplified ML

- Equivalent to ML detector
  - Optimal detection when $\boldsymbol{P}_j$'s are equiprobable

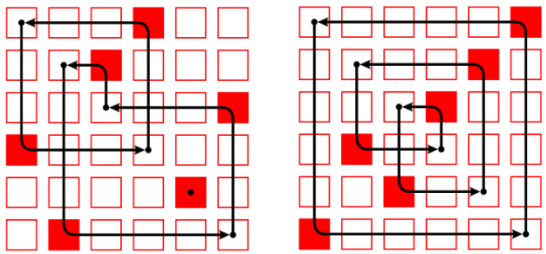
- Computation complexity:
  - Linear in M, N *(number of precoder matrices)*, and C *(size of symbol constellation)*

- Additional step needed to handle ambiguity for 4x4 MIMO with 4 layers

Precoder Detection

Ambiguity in 4x4 MIMO scenario, 4 layers

- 16 4x4 precoder matrices grouped into
  a) Three sets of four precoders
  b) Two sets of two precoders,



where $\boldsymbol{P}_i = \boldsymbol{A}\boldsymbol{P}_j$, $\boldsymbol{P}_i, \boldsymbol{P}_j \in$ a set, $\boldsymbol{A}$ – permutation matrix

- Ambiguity resolved using channel coding built in LTE
  – All precoders in a detected set sent to receive chain
  – The precoder that passes turbo decoding and CRC check is chosen as detected precoder

Precoder Detection – Cluster variance

$$\left\{ \boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{P} \boldsymbol{x}_i + \boldsymbol{n}_i \right\}$$

- Define following objective function for each precoder $\boldsymbol{P}_j$

$$l\left(\boldsymbol{P}_j; \widehat{\boldsymbol{H}}, \boldsymbol{y}, \widehat{\boldsymbol{x}}^{(j)}\right) := \sum_{i=1}^{M} \left\| \widehat{\boldsymbol{x}}_i^{(j)} - \boldsymbol{P}_j^{+} \widetilde{\boldsymbol{y}}_i \right\|^2$$

- Choose the precoder $\boldsymbol{P}_j$ that minimizes $l(\cdot)$

- Similarity with minimum distance decision rule

Precoder Detection – Cluster variance

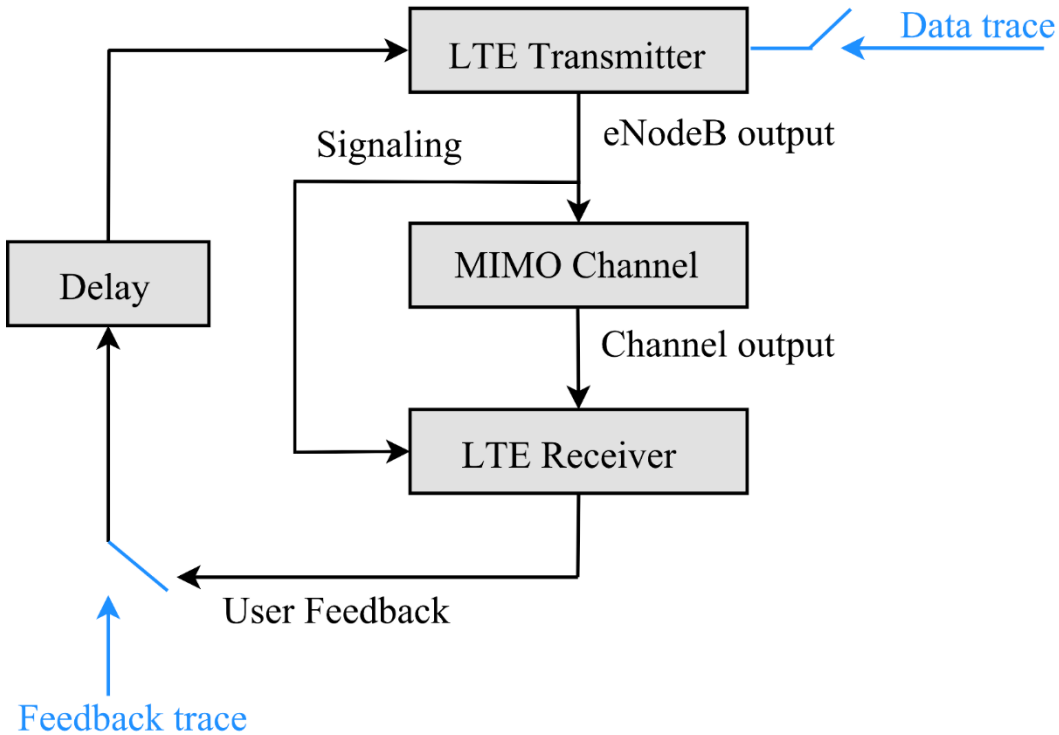
- Not equivalent to ML detector (i.e. not optimal)
  - Modified noise term, after MMSE filter, is **not** i.i.d.

- Computation complexity:
  - Linear in M, N *(number of precoder matrices)*, and C *(size of symbol constellation)*
  - Fewer floating point operations needed compared to Simplified ML algorithm

- Additional step needed to handle ambiguity for 4x4 MIMO
  - Same as with Simplified ML

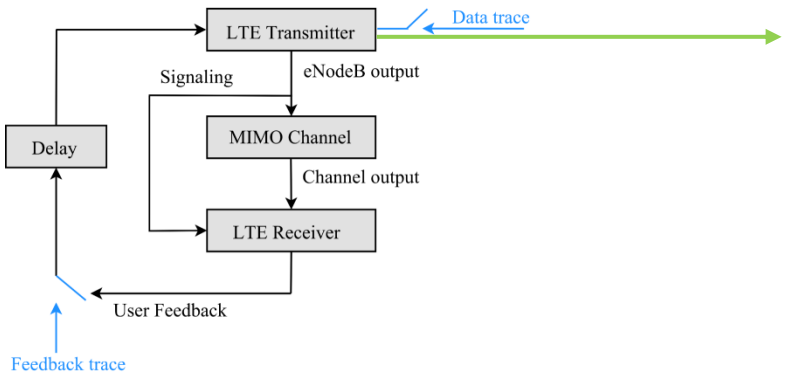
System Model

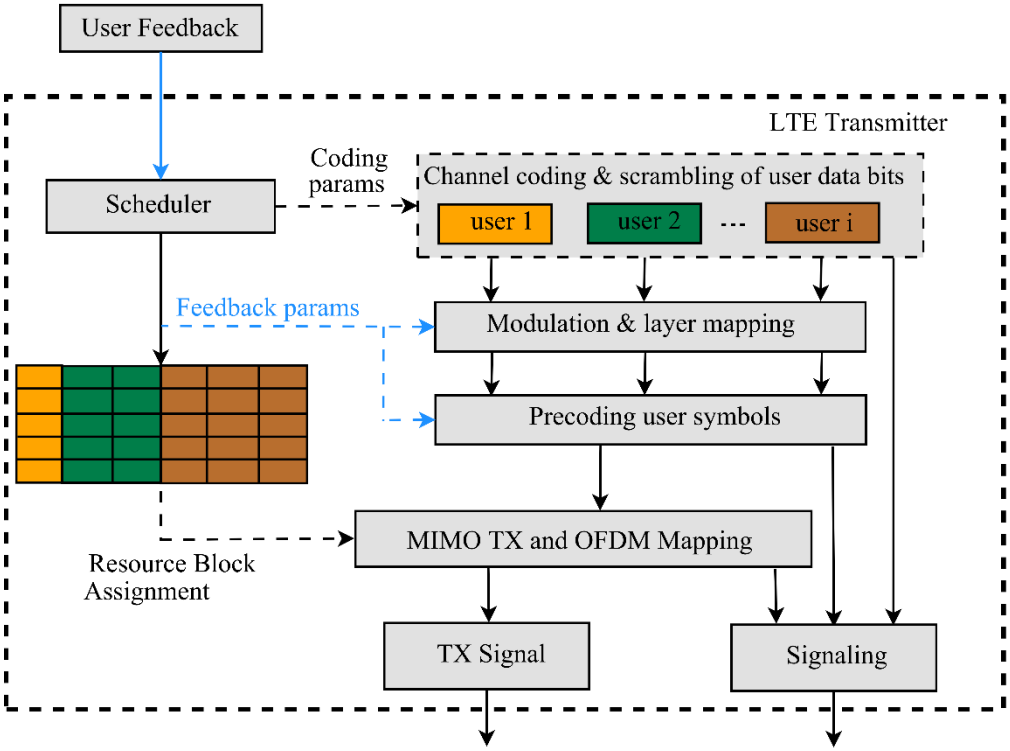
Based on a MATLAB LTE link-level simulator from the Vienna University of Technology



Overall LTE simulator structure

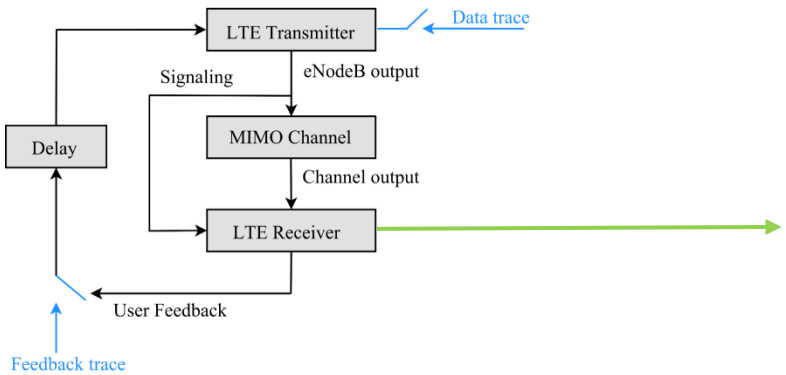
System Model



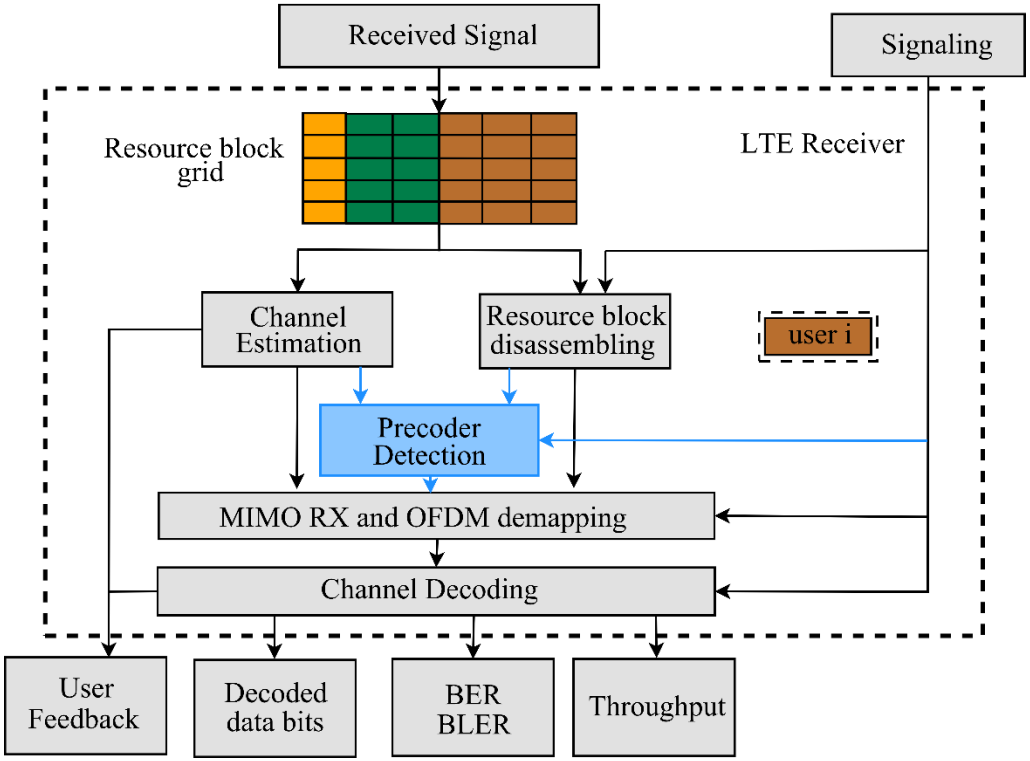


LTE Transmitter

System Model

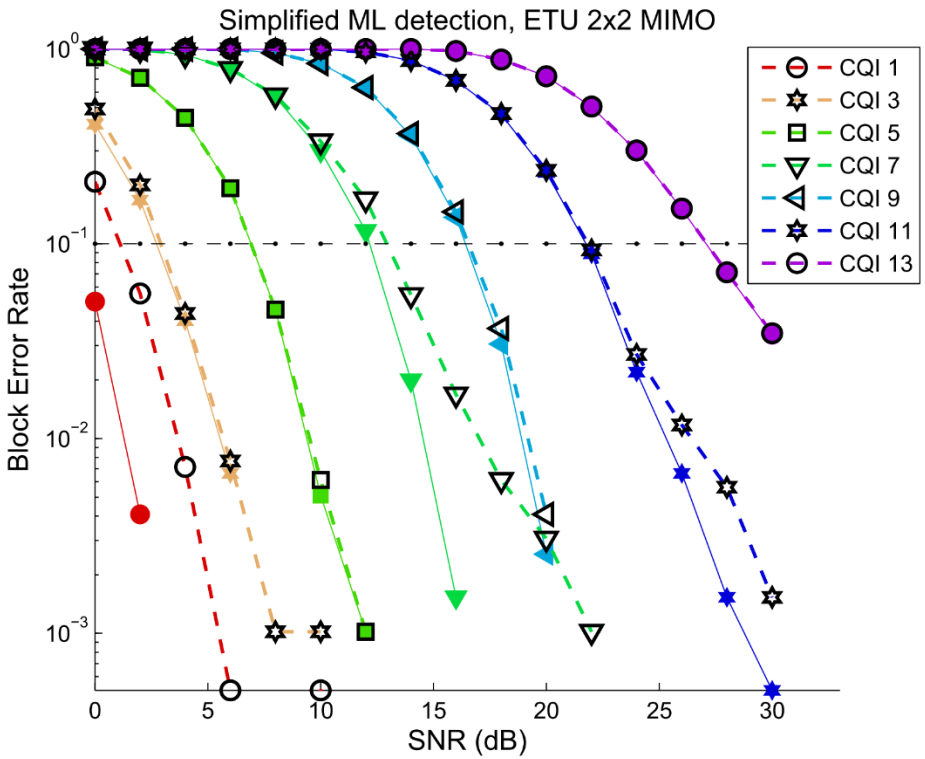


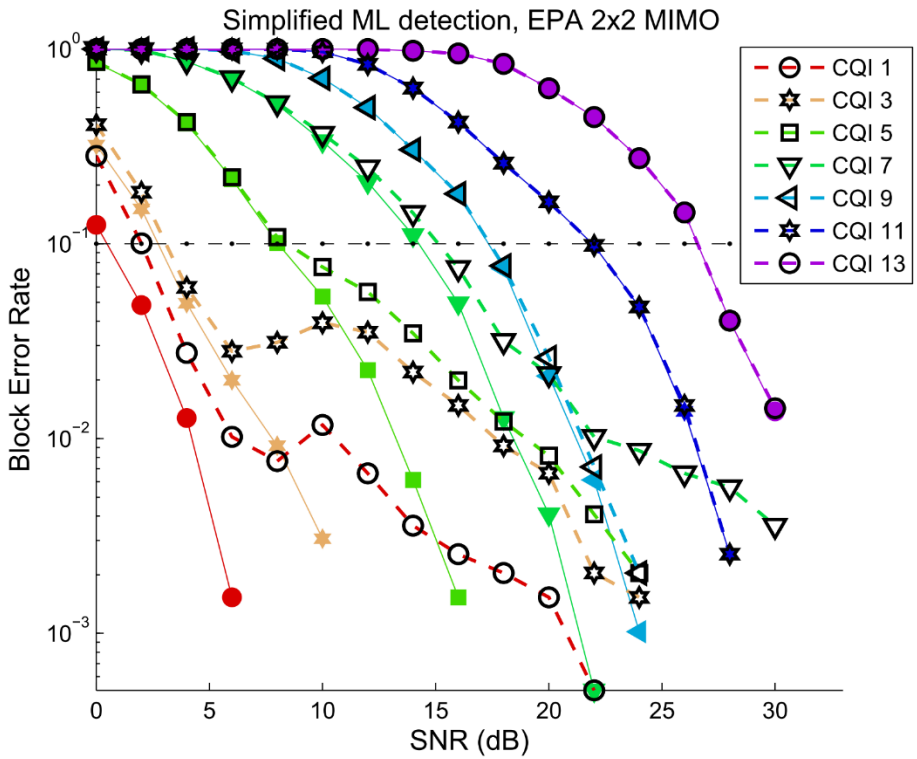
LTE Receiver



Simulation Results – Simplified ML

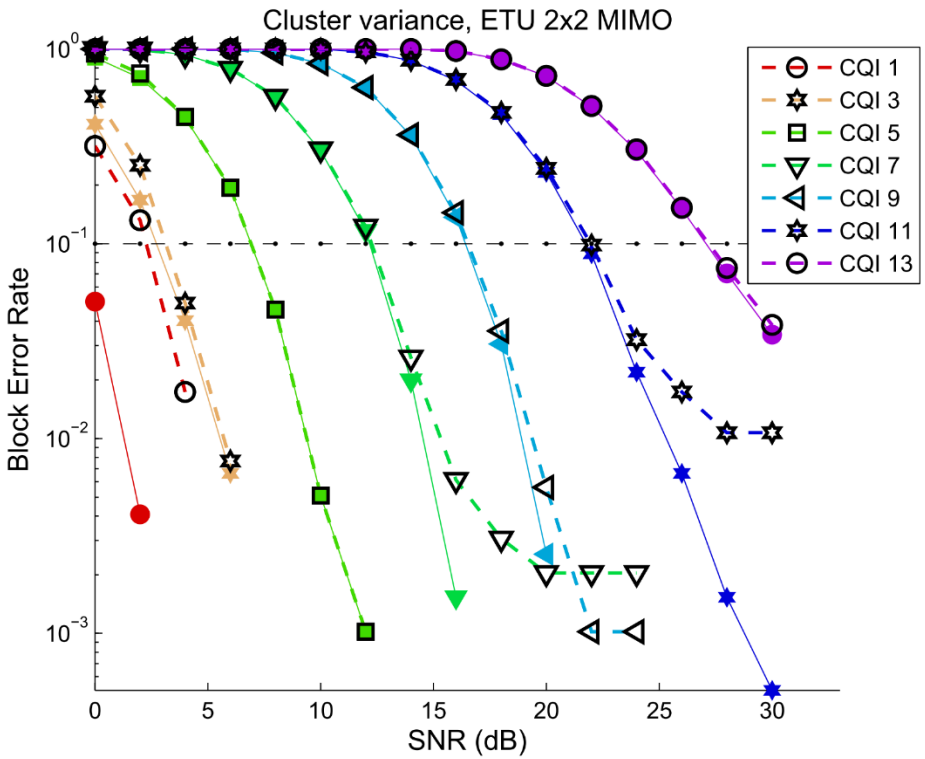
2x2 MIMO: ETU channel at 60 km/h user speed, EPA channel at 3 km/h user speed
Carrier frequency = 2110 MHz, System bandwidth = 1.4 MHz
1000 subframes, 12 RBs scheduled, MMSE channel estimator, SSD receiver

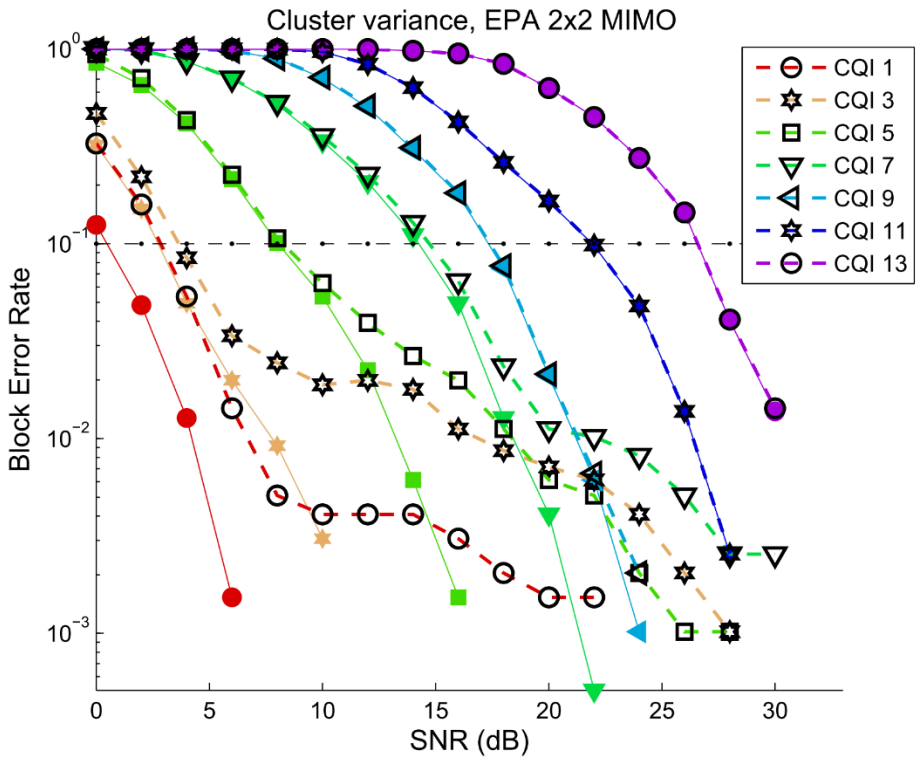




Simulation Results – Cluster variance

2x2 MIMO: ETU channel at 60 km/h user speed, EPA channel at 3 km/h user speed
Carrier frequency = 2110 MHz, System bandwidth = 1.4 MHz
1000 subframes, 12 RBs scheduled, MMSE channel estimator, SSD receiver





Conclusions

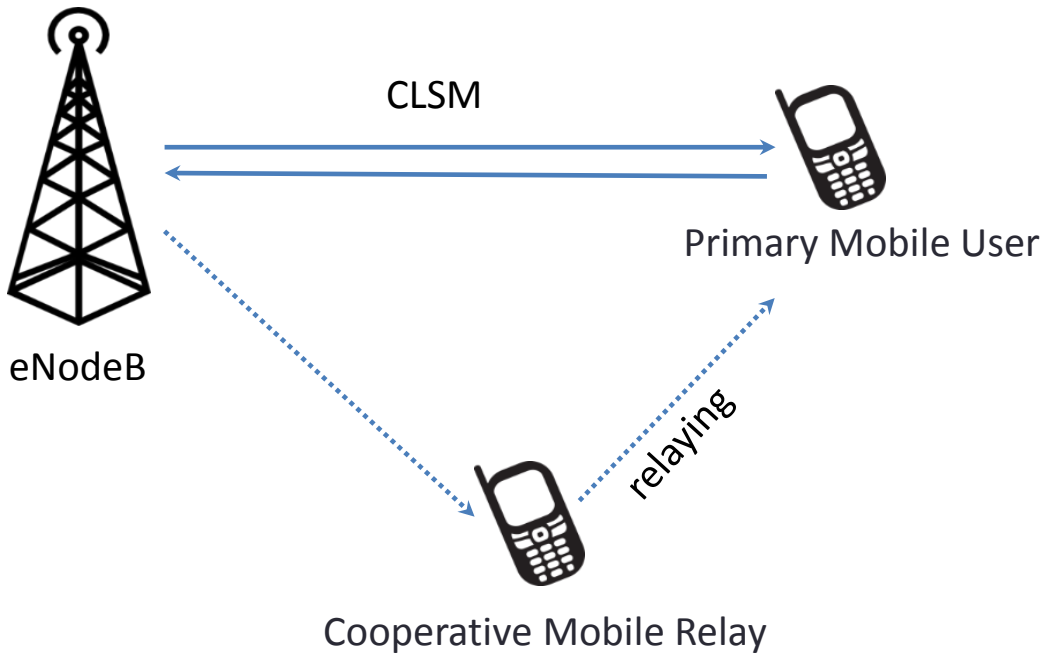
- Simplified ML within 2dB of BLER performance vs. known Precoder Matrix Index (PMI) case
- Simplified ML typically outperforms cluster variance, in 2x2 MIMO
- Performance gap reduced in 4x4 MIMO
  - Additional diversity as each codeword is sent on 2 antennas
- Computational requirement less than SSD receiver (linear in M, # *precoders* N, *constellation size* C)

Future Work

- Consider more realistic simulations
  - Interference from neighboring cells
  - Impact of scheduling multiple users

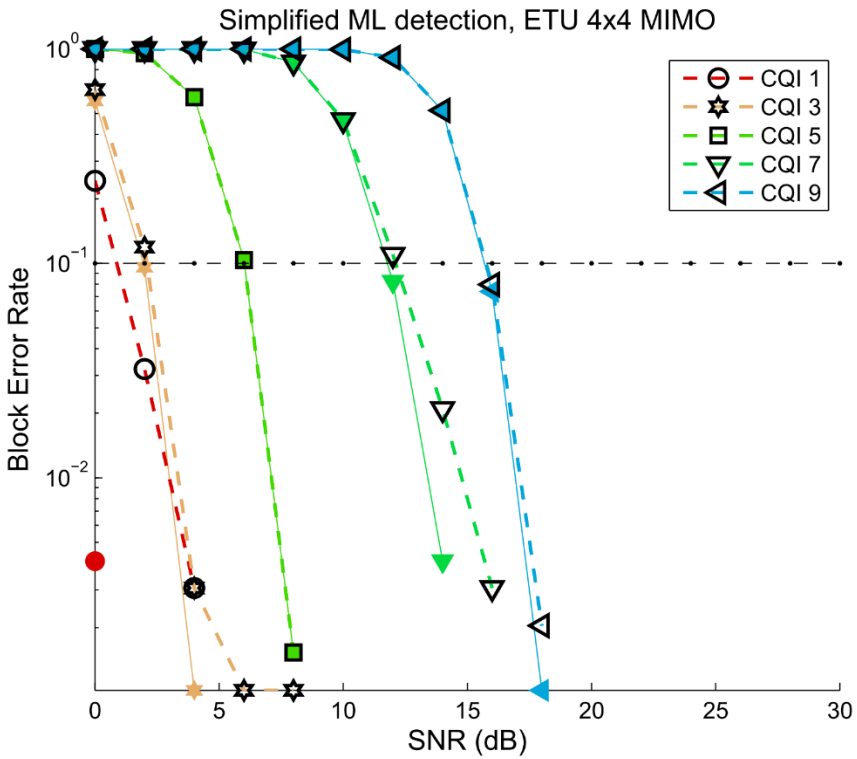
- Conduct experiments using a test-bed
  - Real data and real channels

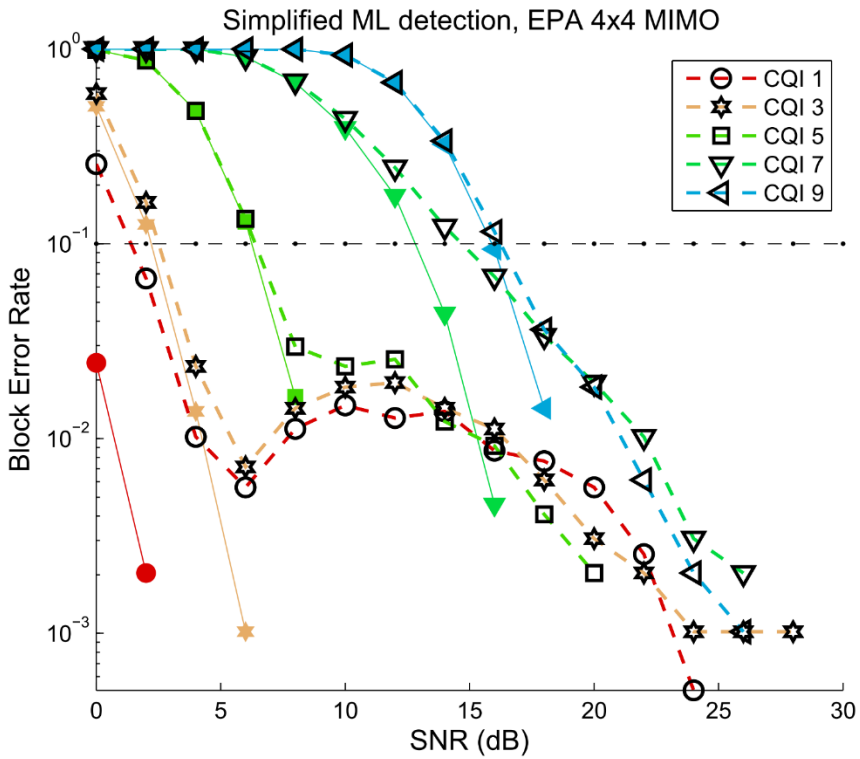
Questions?


CLSM
Primary Mobile User
eNodeB
relaying
Cooperative Mobile Relay

Simulation Results – Simplified ML

4x4 MIMO: ETU channel at 60 km/h user speed, EPA channel at 3 km/h user speed
Carrier frequency = 2110 MHz, System bandwidth = 1.4 MHz
1000 subframes, 12 RBs scheduled, MMSE channel estimator, SSD receiver





Simulation Results – Cluster variance

4x4 MIMO: ETU channel at 60 km/h user speed, EPA channel at 3 km/h user speed
Carrier frequency = 2110 MHz, System bandwidth = 1.4 MHz
1000 subframes, 12 RBs scheduled, MMSE channel estimator, SSD receiver