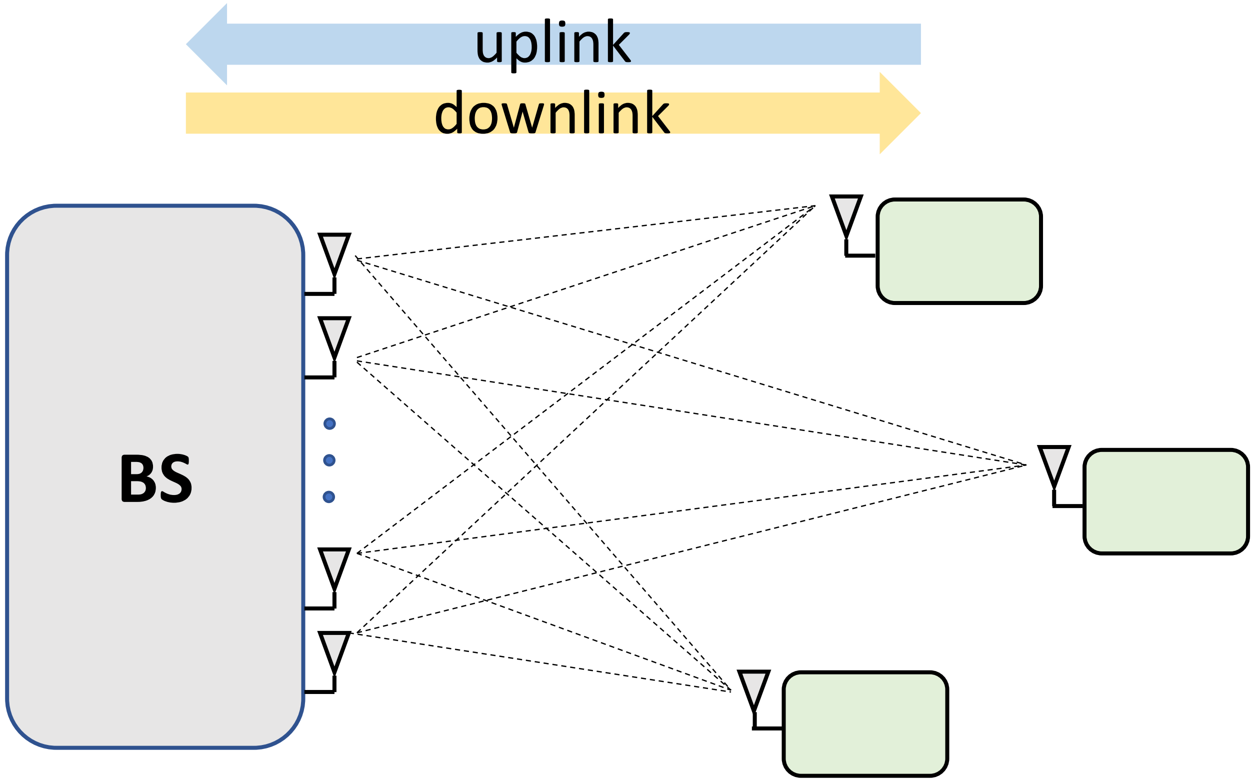
Design Trade-offs for Decentralized Baseband Processing in Massive MU-MIMO Systems

Kaipeng Li, James McNaney, Oscar Castañeda, **Chance Tarver**, Charles Jeon, Joseph Cavallaro, Christoph Studer

Asilomar Conference on Signals, Systems, and Computers
November 5, 2019

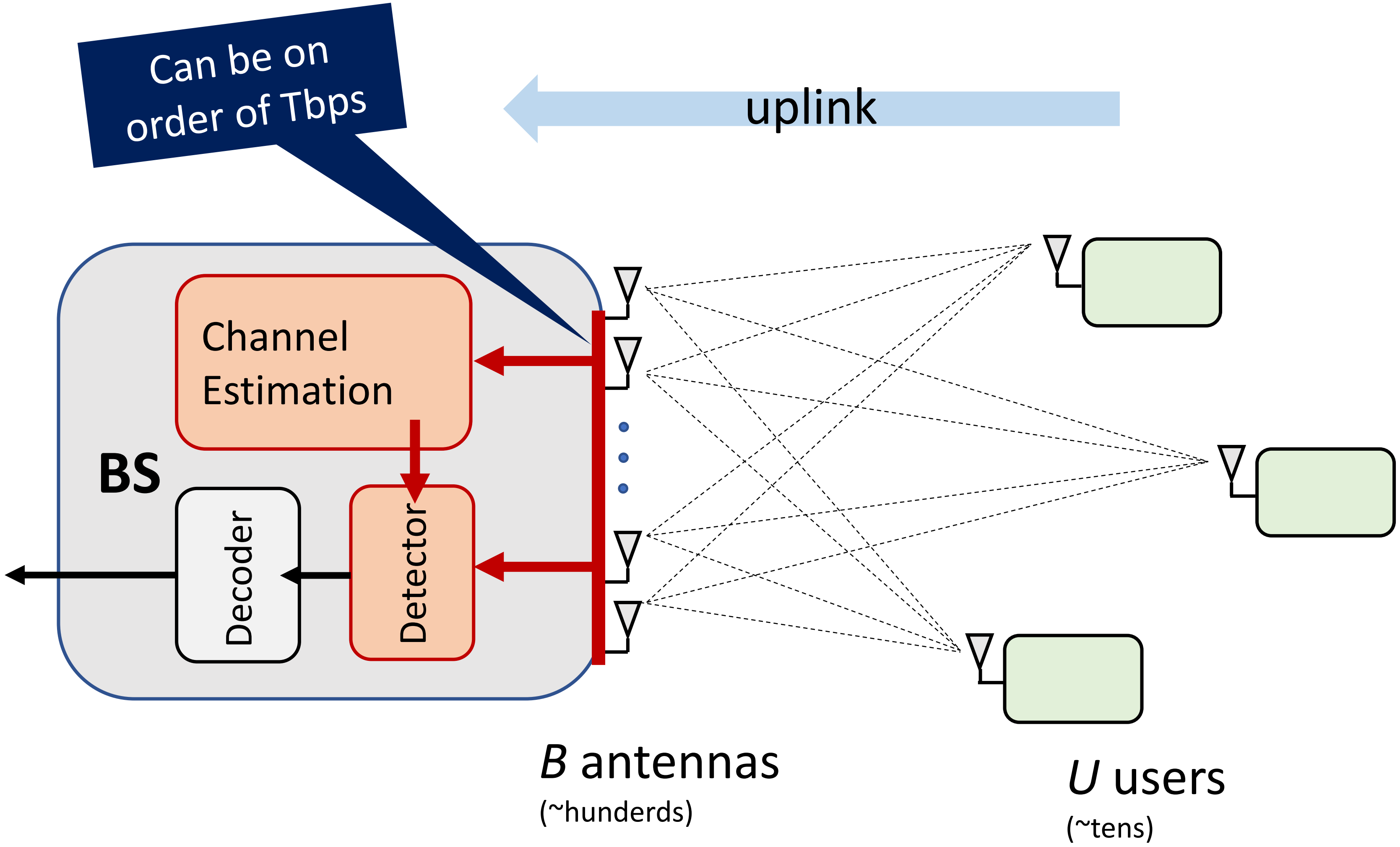
Massive MU-MIMO systems



B antennas
(~hunderds)

U users
(~tens)

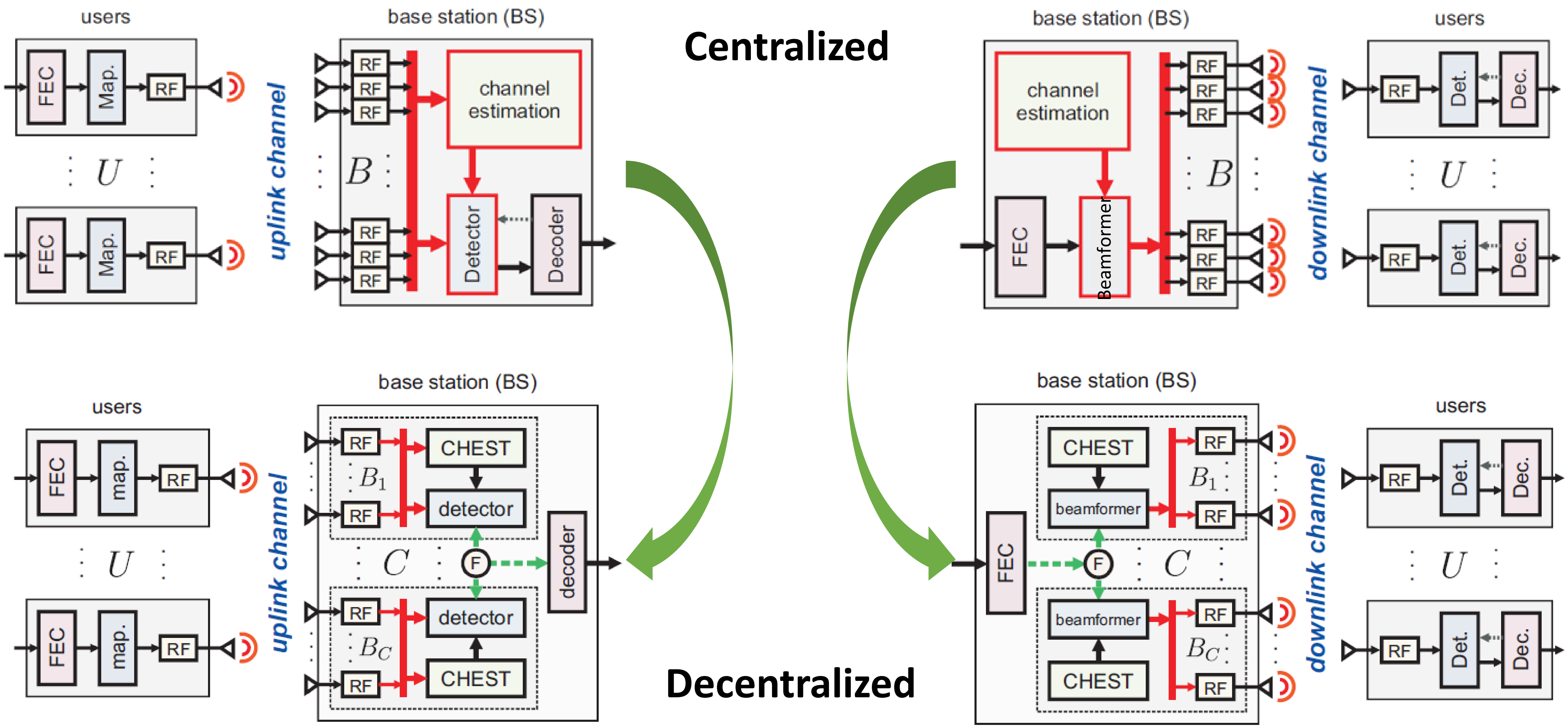
How do we handle this much data?



Possible Limitations:

- Chip I/O and interconnection bandwidth
- On-chip memory and storage
- Computing capability of modern computing fabrics

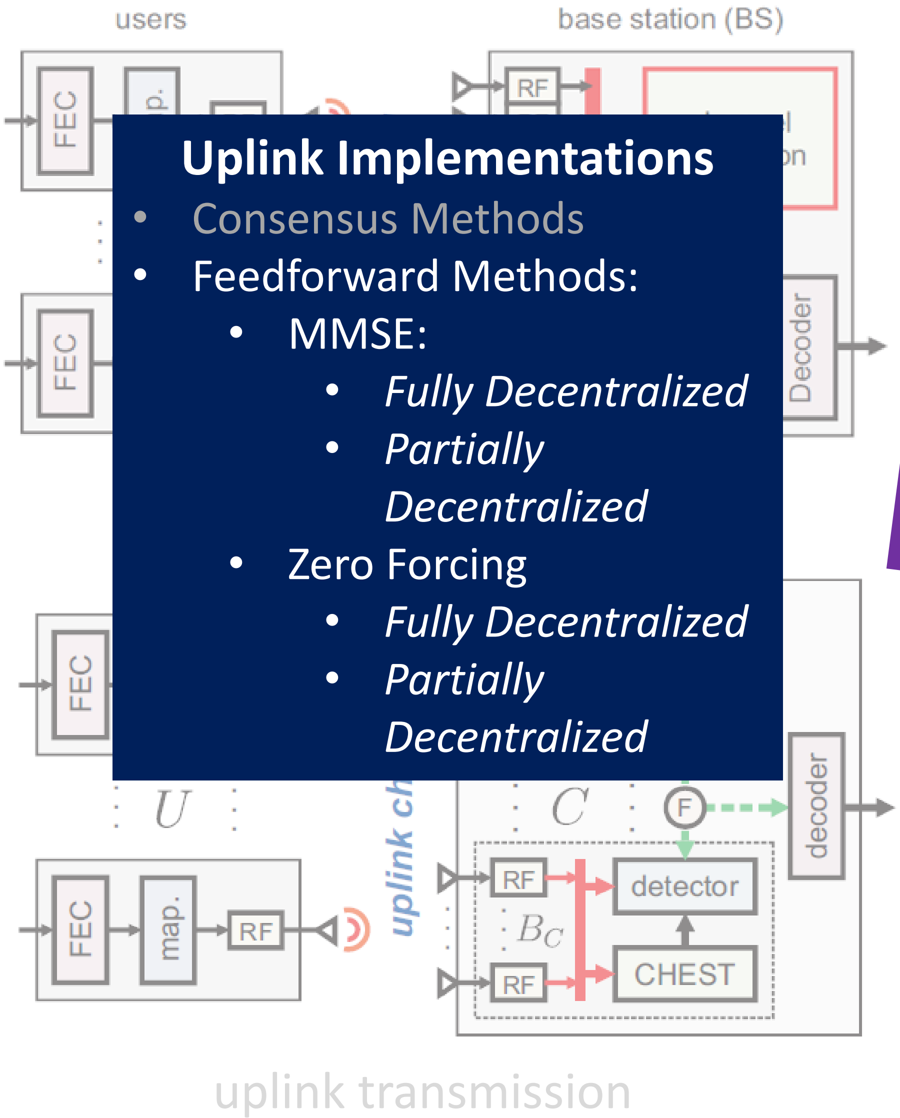
Decentralized to resolve bottlenecks



uplink transmission

downlink transmission

Decentralized to resolve bottlenecks



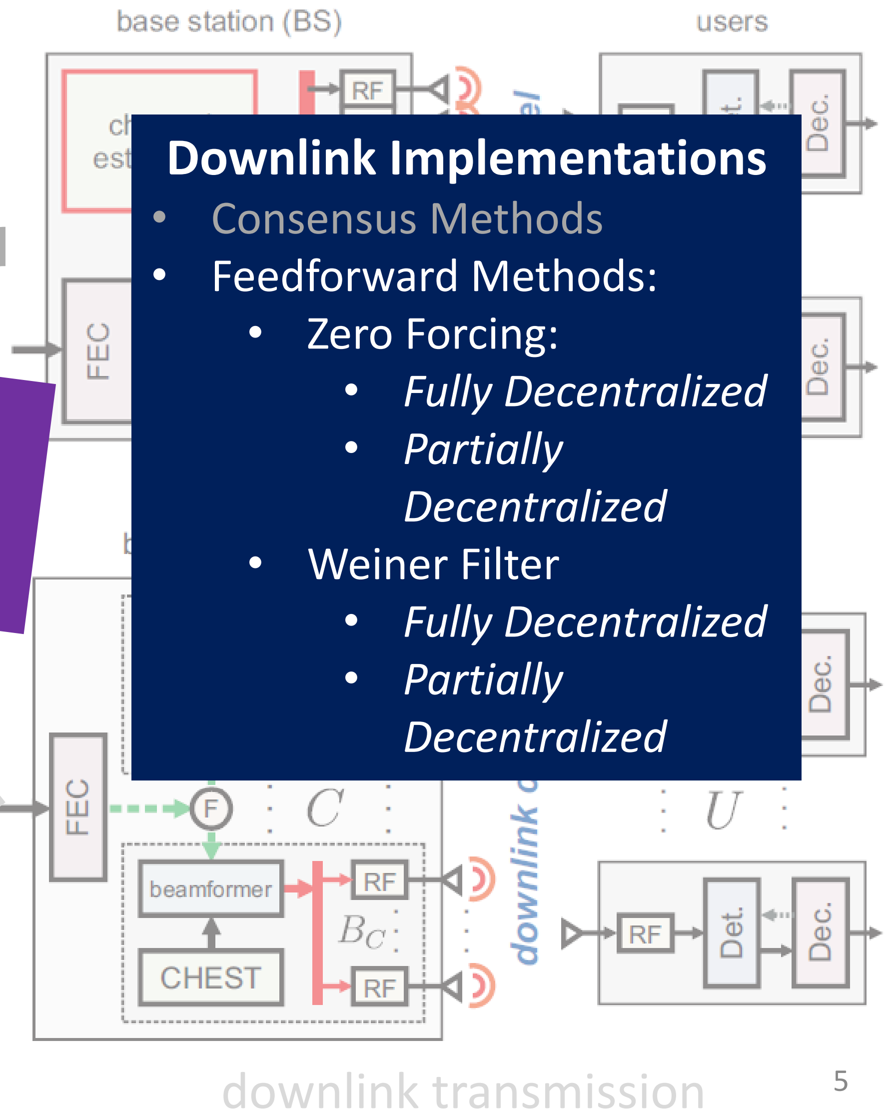
Uplink Implementations

- Consensus Methods
- Feedforward Methods:
  - MMSE:
    - *Fully Decentralized*
    - *Partially Decentralized*
  - Zero Forcing
    - *Fully Decentralized*
    - *Partially Decentralized*

Centralized

How should we pick an implementation?

Decentralized



Downlink Implementations

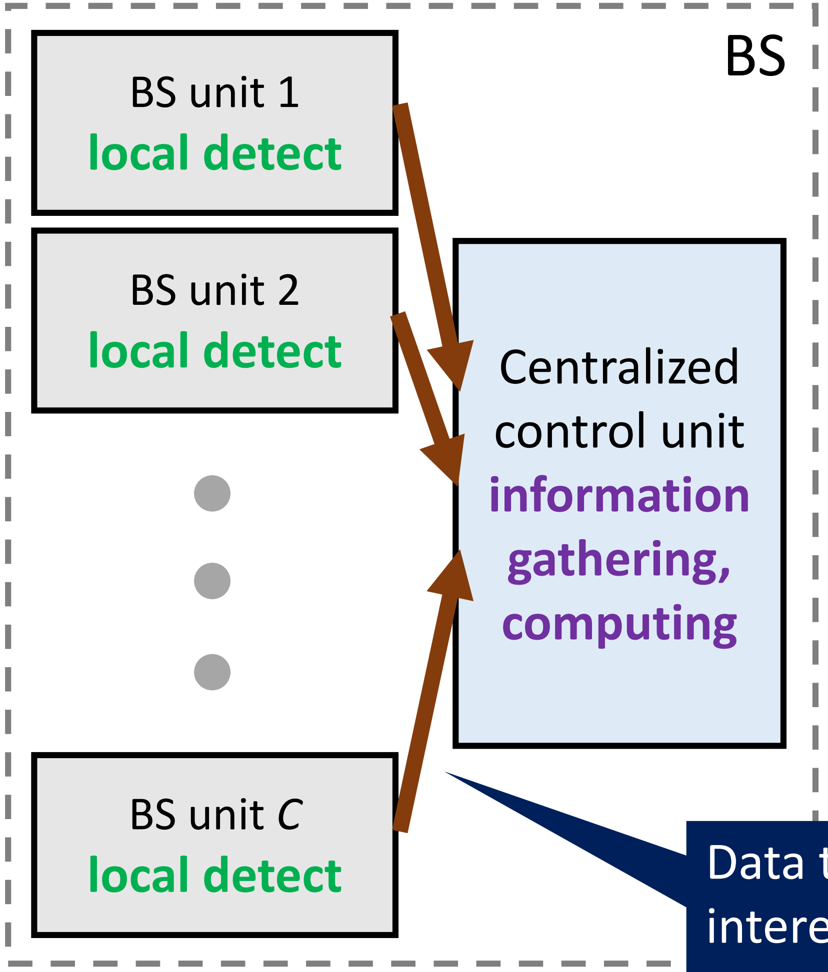
- Consensus Methods
- Feedforward Methods:
  - Zero Forcing:
    - *Fully Decentralized*
    - *Partially Decentralized*
  - Weiner Filter
    - *Fully Decentralized*
    - *Partially Decentralized*

Outline

- Overview of decentralized architectures and algorithms
- Architecture trade-offs
- Algorithm trade-offs
- Precision trade-offs
- Practical design flow
- Conclusion

Decentralized feedforward architecture

Feedforward local information ***only once*** instead of multiple rounds to centralized unit



Data transfer we are interested in. Would be on PCIe, NVLink, InfiniBand.

Example: uplink system

Partially decentralized (PD) architecture:
less local computation + more centralized computation

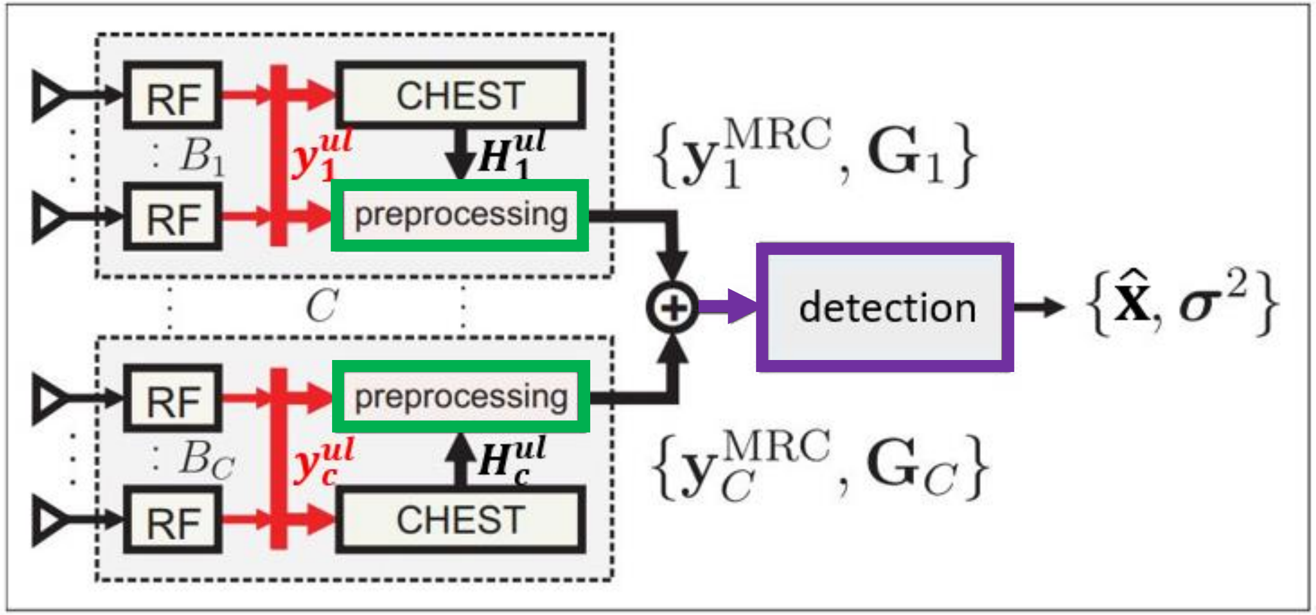
Fully decentralized (FD) architecture:
more local computation + less centralized computation

Uplink linear MMSE detection

Centralized linear MMSE (C-LMMSE) detection:

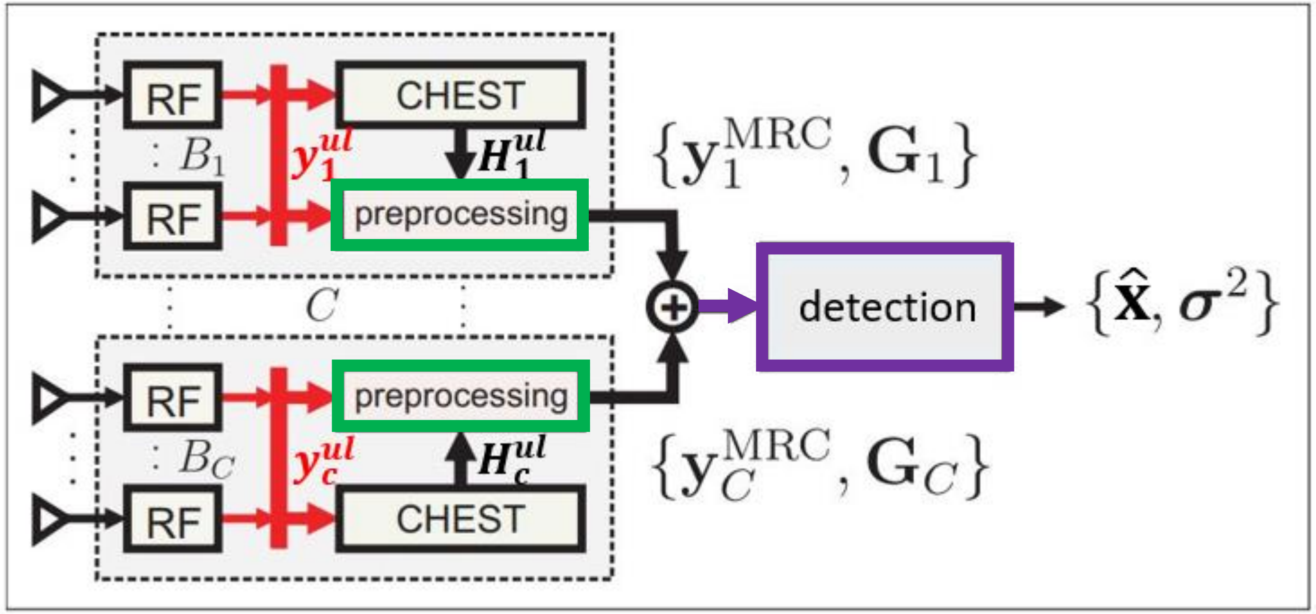
$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^H\boldsymbol{H} + \frac{N_0}{E_x}\boldsymbol{I})^{-1}\,\boldsymbol{H}^H\boldsymbol{y}$$
$$= (\ \boldsymbol{G}\ + \frac{N_0}{E_x}\boldsymbol{I})^{-1}\boldsymbol{y}^{\mathrm{MRC}}$$

Partially decentralization: ***decentralized*** matrix preprocessing + ***centralized*** detection



$\boldsymbol{\sigma}^2$: error variance

Uplink linear MMSE detection



Centralized LMMSE (C-LMMSE):

$$\boldsymbol{G} = \boldsymbol{H}^H\boldsymbol{H} \qquad \boldsymbol{y}^{\mathrm{MRC}} = \boldsymbol{H}^H\boldsymbol{y}$$

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{G} + \frac{N_0}{E_x}\boldsymbol{I}\right)^{-1}\boldsymbol{y}^{\mathrm{MRC}}$$

Partially Decentralized:

$$\boldsymbol{G}_c = \boldsymbol{H}_c^H\boldsymbol{H}_c \qquad \boldsymbol{y}_c^{\mathrm{MRC}} = \boldsymbol{H}_c^H\boldsymbol{y}_c$$

$$\boldsymbol{G} = \sum_{c=1}^{C}\boldsymbol{G}_c \qquad \boldsymbol{y}^{\mathrm{MRC}} = \sum_{c=1}^{C}\boldsymbol{y}_c^{\mathrm{MRC}}$$

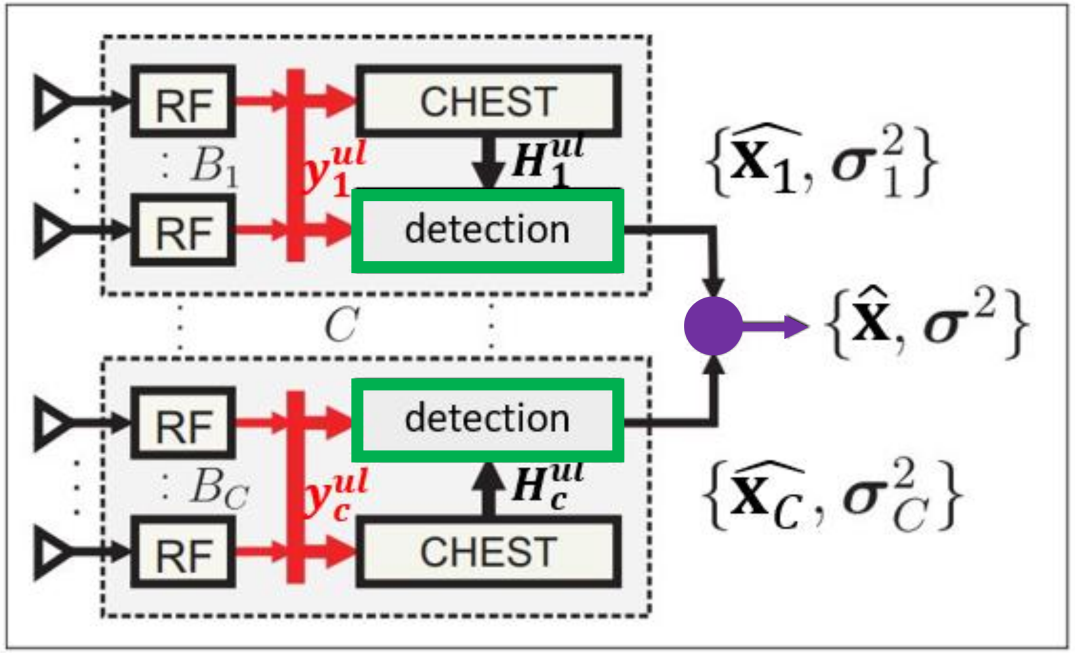
$$\hat{\boldsymbol{x}} = \left(\boldsymbol{G} + \frac{N_0}{E_x}\boldsymbol{I}\right)^{-1}\boldsymbol{y}^{\mathrm{MRC}}$$

PD-LMMSE obtains *the same* $\hat{\boldsymbol{x}}$ as C-LMMSE

Complexity: $O(B_cU^2)$ + $O(U^3)$ mults.

Data transfer size: **$O(U^2)$** samples / cluster

Fully decentralized (FD-) LMMSE detection



Decentralized local detection

$$\hat{\boldsymbol{x}}_c = (\boldsymbol{G}_c + \frac{N_0}{E_x}\boldsymbol{I})^{-1}\boldsymbol{y}_c^{\mathrm{MRC}}$$

Fusion of local $\hat{\boldsymbol{x}}_c$ using weights, $\boldsymbol{\lambda}_c$:

$$\hat{\boldsymbol{x}} = \sum_{c=1}^{C} \boldsymbol{\lambda}_c \hat{\boldsymbol{x}}_c$$

Optimal $\boldsymbol{\lambda}_c$ is a function of σ_c

Complexity: **$O(B_c U^2)$** + **$O(U^3)$** mults.

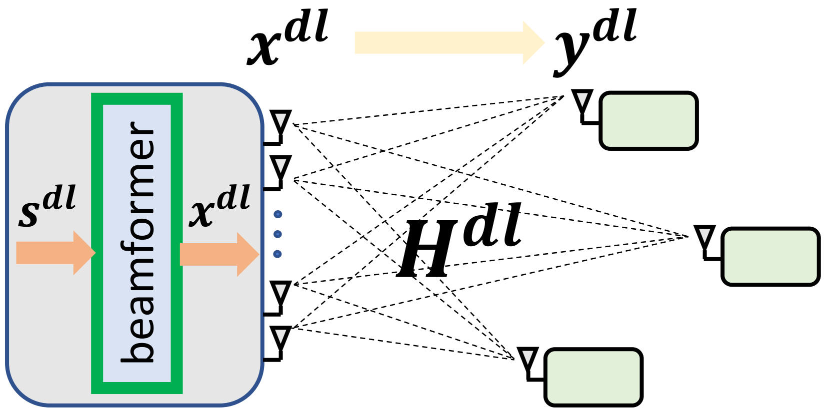
Data transfer size: **O(U)** samples / cluster

Downlink Beamforming

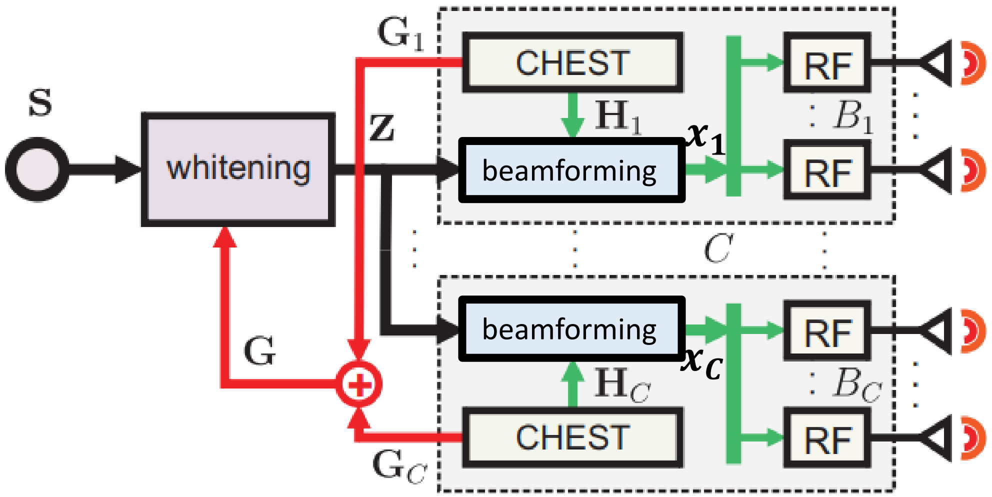
- **Linear beamforming:**
  - Power constraint: $\mathbf{E}[\|\boldsymbol{x}\|^2] \leq \rho^2$
  - Precoding matrix: $\boldsymbol{P}$
  - Linear precoding: $\boldsymbol{x} = \boldsymbol{Ps}$

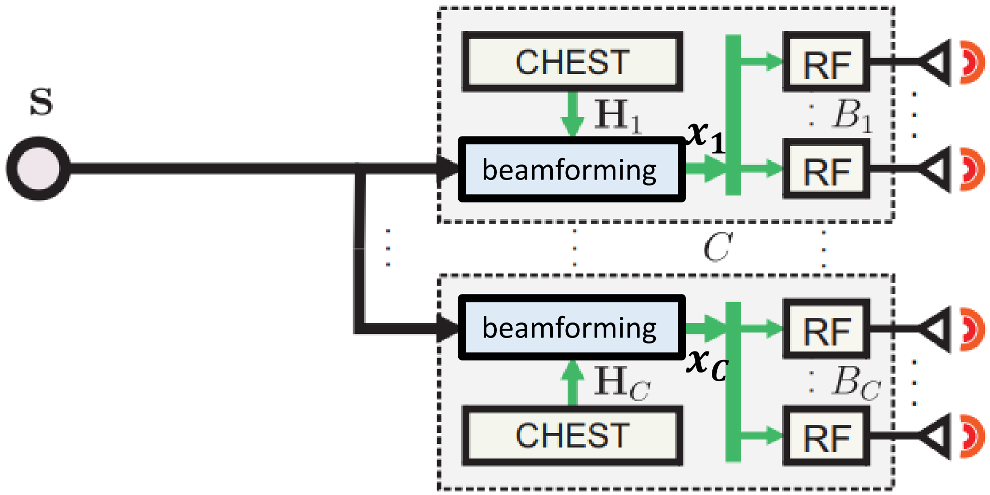
- **Zero-Forcing beamforming:**
  - Precoding Matrix: $\boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{H}^H)^{-1} = \boldsymbol{H}^H\boldsymbol{G}^{-1}$
  - Power constraint: $\widehat{\boldsymbol{x}} = \rho\|\widehat{\boldsymbol{x}}\|_2$

- **Channel reciprocity:**
  - TDD Transmission: $\boldsymbol{H}^{dl} = (\boldsymbol{H}^{ul})^H$



Decentralized feedforward ZF beamforming





Partially decentralized ZF beamforming:

Set $\rho_c^2 = \rho^2/C$

$\boldsymbol{G}_c = \boldsymbol{H}_c\boldsymbol{H}_c^H$

$\boldsymbol{G} = \sum_{c=1}^{C} \boldsymbol{G}_c \quad \boldsymbol{z} = \boldsymbol{G}^{-1}\boldsymbol{s}$

Broadcast $\boldsymbol{z}$ to local clusters

$\hat{\boldsymbol{x}}_c = \boldsymbol{H}_c^H\boldsymbol{z}, \qquad \hat{\boldsymbol{x}}_c = \rho_c \|\hat{\boldsymbol{x}}_c\|_2$

Complexity: $O(B_cU^2)$ + $O(U^3)$ mults.

Data transfer: **$O(U^2)$** samples / cluster

Fully decentralized ZF beamforming:

Broadcast $\boldsymbol{s}$ and set $\rho_c^2 = \rho^2/C$

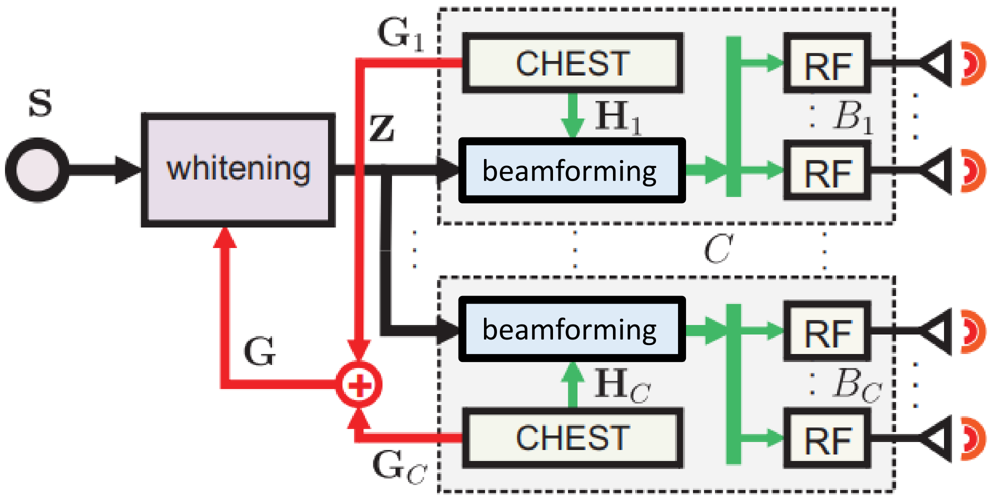
$\hat{\boldsymbol{x}}_c = \boldsymbol{H}_c^H(\boldsymbol{H}_c\boldsymbol{H}_c^H)^{-1}\boldsymbol{s}$

$\hat{\boldsymbol{x}}_c = \rho_c \|\hat{\boldsymbol{x}}_c\|_2$

Complexity: $O(B_cU^2)$ + $O(U^3)$ mults.

Data transfer: **$O(U)$** samples / cluster

Decentralized feedforward Wiener Filter (WF) beamforming



Partially decentralized WF beamforming

Set $\rho_c^2 = \rho^2/C$

$\boldsymbol{G}_c = \boldsymbol{H}_c \boldsymbol{H}_c^H$

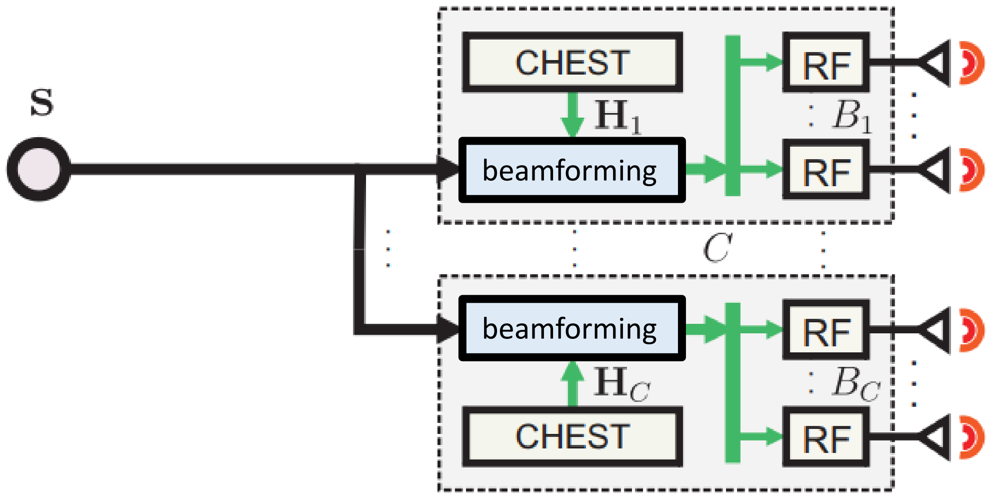
$\boldsymbol{G} = \sum_{c=1}^{C} \boldsymbol{G}_c \quad \boldsymbol{z} = \frac{1}{\beta}(\boldsymbol{G} + \gamma \boldsymbol{I}_U)^{-1} \boldsymbol{s}$

Broadcast $\boldsymbol{z}$ to local BS unit

$\hat{\boldsymbol{x}}_c = \boldsymbol{H}_c^H \boldsymbol{z}$

Complexity: $O(B_c U^2) + O(U^3) + O(\beta)$ mult.

Data transfer: **$O(U^2)$** samples / cluster



Fully decentralized WF beamforming

Broadcast $\boldsymbol{s}$ and set $\rho_c^2 = \rho^2/C$

$\boldsymbol{P}_c = \frac{1}{\beta_c} \boldsymbol{H}_c^H (\boldsymbol{H}_c \boldsymbol{H}_c^H + \gamma \boldsymbol{I}_U)^{-1} \boldsymbol{s}$

$\hat{\boldsymbol{x}}_c = \boldsymbol{P}_c \boldsymbol{s}$

Complexity: $O(B_c U^2) + O(U^3) + O(\beta)$ mult.

Data transfer: **$O(U)$** samples / cluster

Architecture Trade-offs: PD vs. FD

PD-MMSE and FD-MMSE: Data transfer Depends on channel coherency

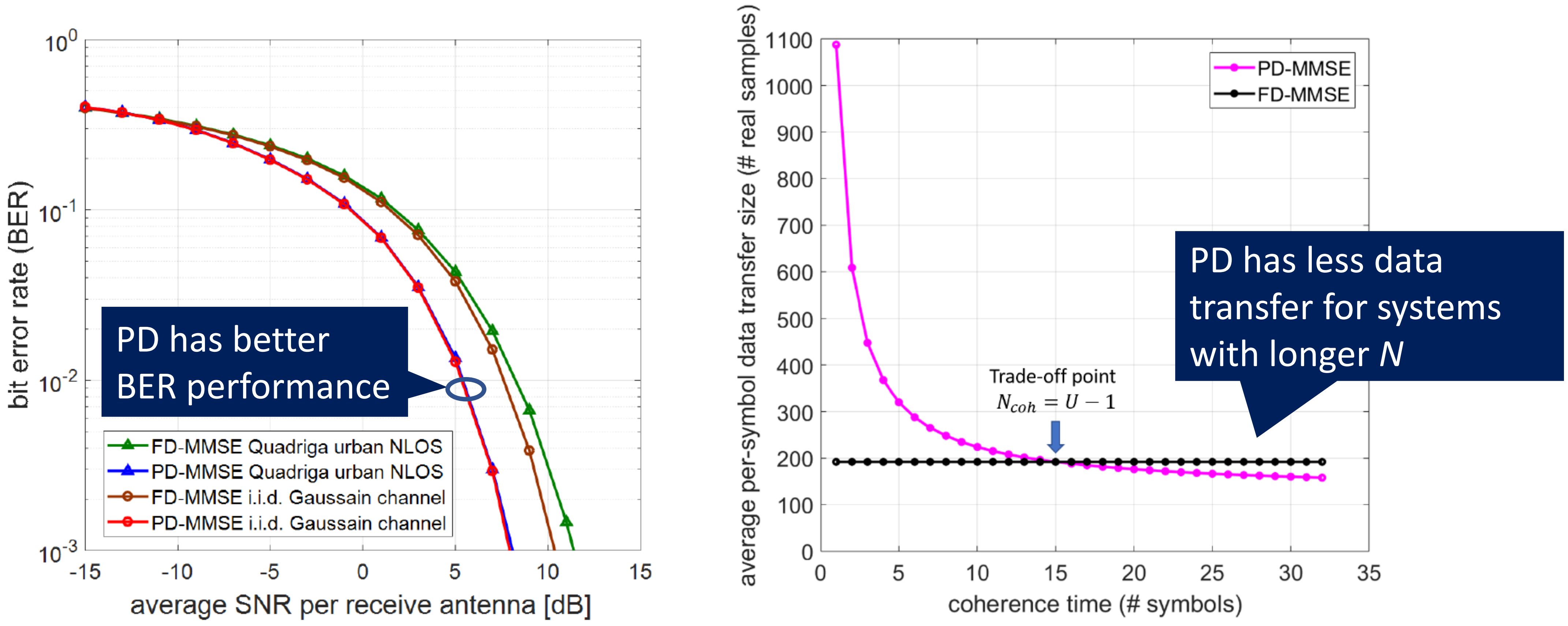
- BER: Centralized MMSE = PD-MMSE, FD-MMSE sacrifices BER
- Computation (timing) complexity: PD-MMSE = FD-MMSE
- N_{coh} : Period in which we update channel state information

$$m_{PD} = \frac{C \times (U^2 - U + 2N_{coh}U)}{N_{coh}}. \quad m_{FD} = \frac{C \times 3N_{coh}U}{N_{coh}} = 3CU.$$

PD vs. FD trade-off on BER and data transfer

C=4, *U*=16, B_c=32, *B*=128, 16QAM

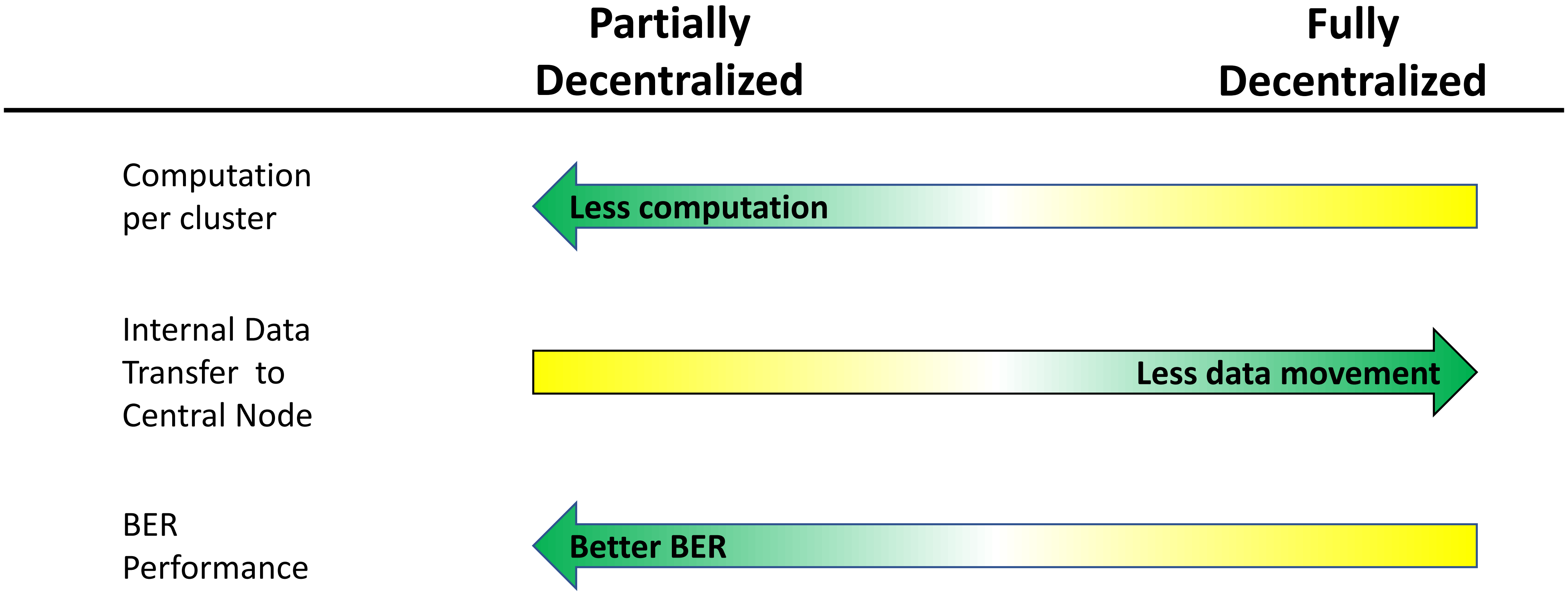
Simple i.i.d. Gaussian channel and Quadriga NLOS urban campus channel



(a) BER: PD-MMSE vs. FD-MMSE (b) Data transfer size vs. N_{coh}

Decentralized feedforward architecture

Feedforward local information ***only once*** instead of multiple rounds to centralized unit



Algorithm Trade-offs: Explicit vs. Implicit method

- Example: PD-MMSE with explicit matrix inversion vs. implicit matrix inversion
- Implicit matrix inversion $\boldsymbol{A}^{-1} = (\boldsymbol{G} + \frac{N_0}{E_x}\boldsymbol{I})^{-1}$ for PD-MMSE
  - $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{H}}$ (Cholesky decomposition, $\boldsymbol{L}$ is lower triangular matrix)
  - Get $\boldsymbol{z}$ by solving $\boldsymbol{L}\boldsymbol{z} = \boldsymbol{y}^{\boldsymbol{MRC}}$ using forward substitution
  - Get $\hat{\boldsymbol{x}}$ by solving $\boldsymbol{L}^{\boldsymbol{H}}\hat{\boldsymbol{x}} = \boldsymbol{z}$ using backward substitution
- Per-symbol complexity of explicit and implicit methods depend on N_{coh}

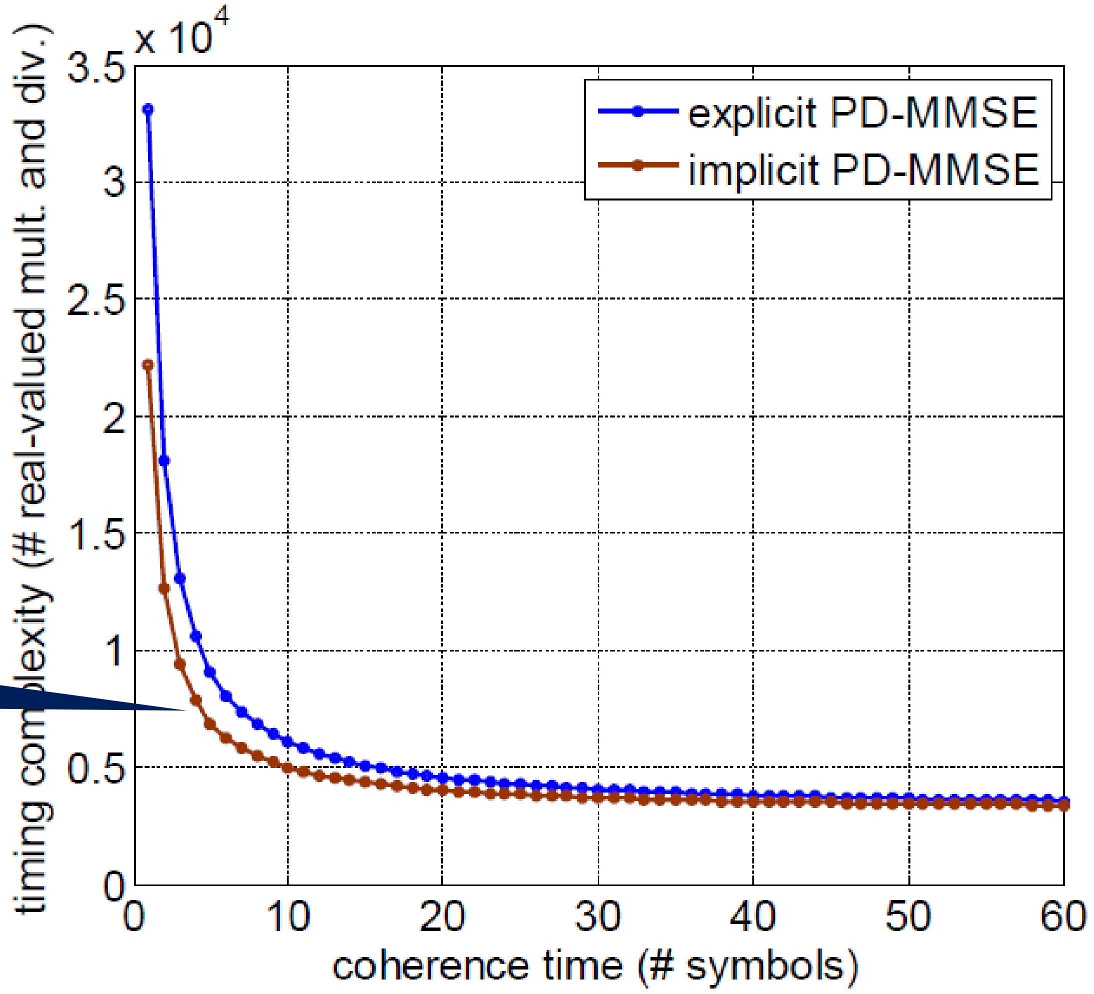
$$n_{ex} = (2B_cU^2 + \frac{10}{3}U^3 - \frac{1}{3}U)/N_{coh} + 4B_cU + 4U^2$$

$$n_{im} = (2B_cU^2 + \frac{2}{3}U^3 + \frac{1}{3}U)/N_{coh} + 4B_cU + 4U^2$$

Complexity of explicit vs. implicit PD-MMSE

C=4, U=16, B_c=32, B=128

Implicit always has lower complexity



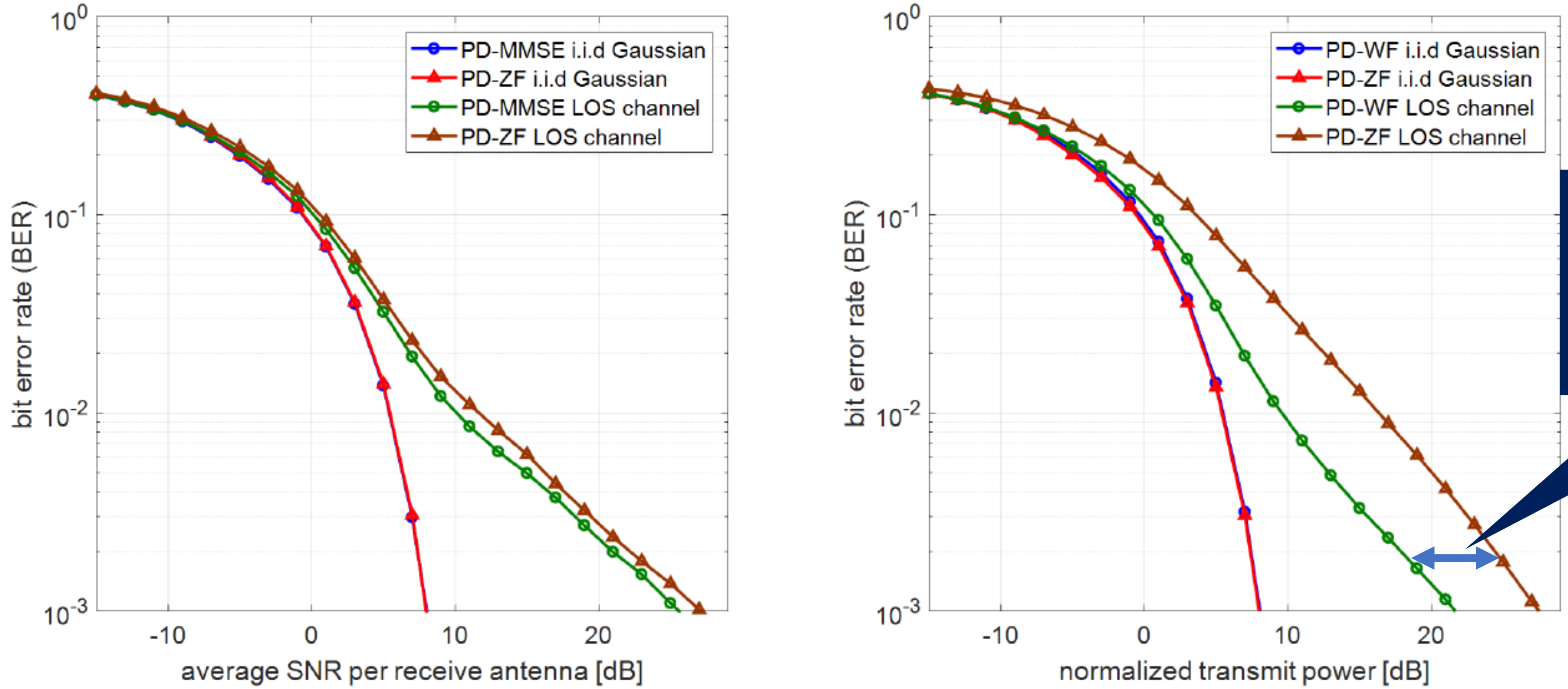
Reusing Uplink (UL) Results for Downlink (DL)

- Channel reciprocity in TDD system: $\boldsymbol{H}^{\boldsymbol{UL}} = (\boldsymbol{H}^{\boldsymbol{DL}})^{\boldsymbol{H}}$
- Gram matrix: $\boldsymbol{G}^{\boldsymbol{DL}} = \boldsymbol{H}^{\boldsymbol{DL}}(\boldsymbol{H}^{\boldsymbol{DL}})^{\boldsymbol{H}} = (\boldsymbol{H}^{\boldsymbol{UL}})^{\boldsymbol{H}}\boldsymbol{H}^{\boldsymbol{UL}} = \boldsymbol{G}^{\boldsymbol{UL}}$
- Store and reuse computed uplink results for downlink to reduce complexity
- UL MMSE detection + DL WF beamforming can only reuse $\boldsymbol{G}^{\boldsymbol{UL}}$
- UL ZF detection + DL ZF beamforming can even reuse $(\boldsymbol{G}^{\boldsymbol{UL}})^{-\mathbf{1}}$

UL and DL integration trade-offs on BER and complexity

Example: UL *PD-MMSE + DL PD-WF* integration vs. *UL ZF + DL ZF* integration

C=4, U=16, B_c=32, B=128, 16QAM, LOS channel



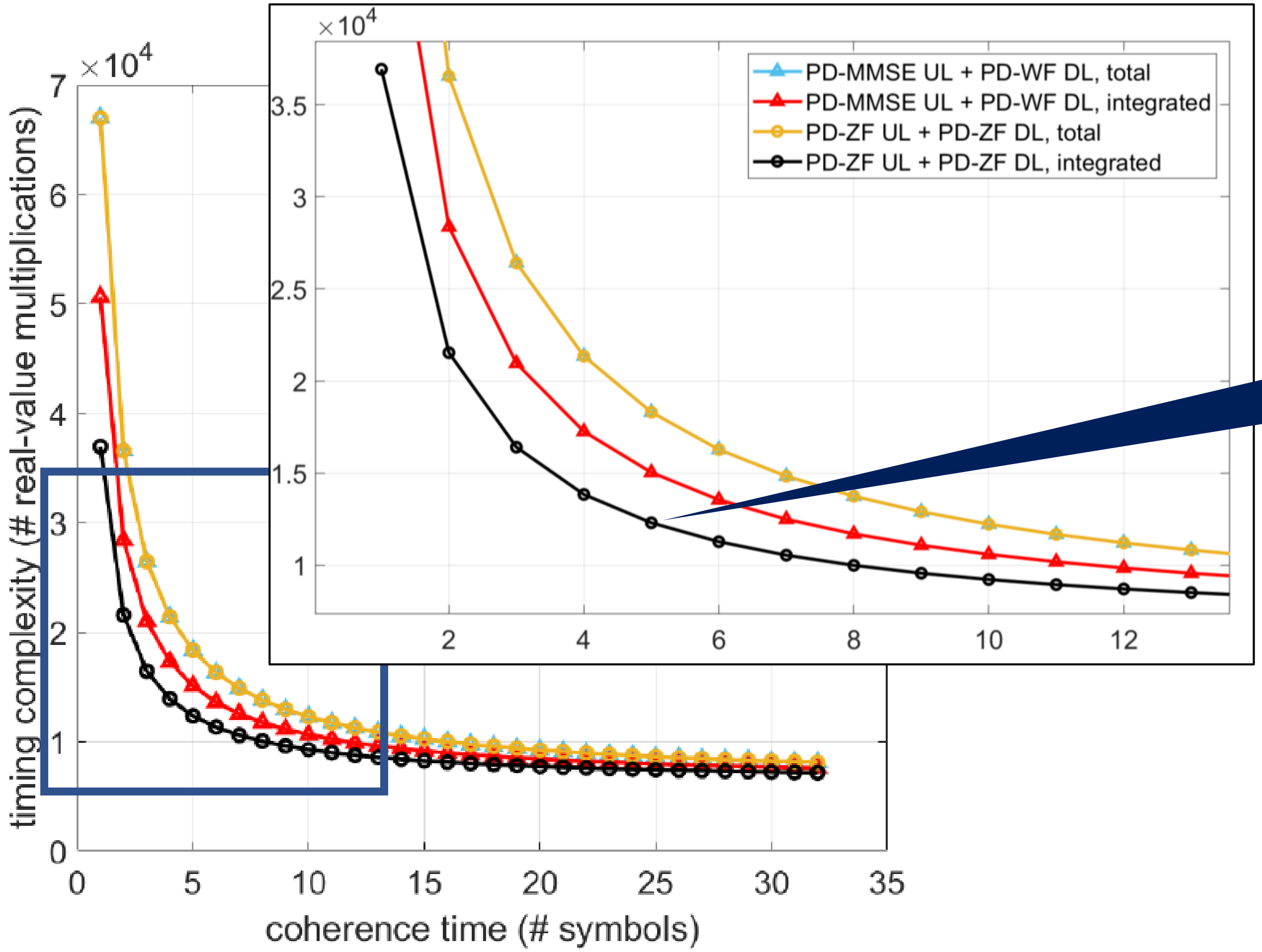
(a) BER: PD-MMSE detection vs. PD-ZF detection (b) BER: PD-WF precoding vs. PD-ZF precoding

MMSE and WF offer better performance

UL and DL integration trade-offs on BER and complexity

Example: UL *PD-MMSE + DL PD-WF* integration vs. *UL ZF + DL ZF* integration

C=4, U=16, B_c=32, B=128, 16QAM, LOS channel

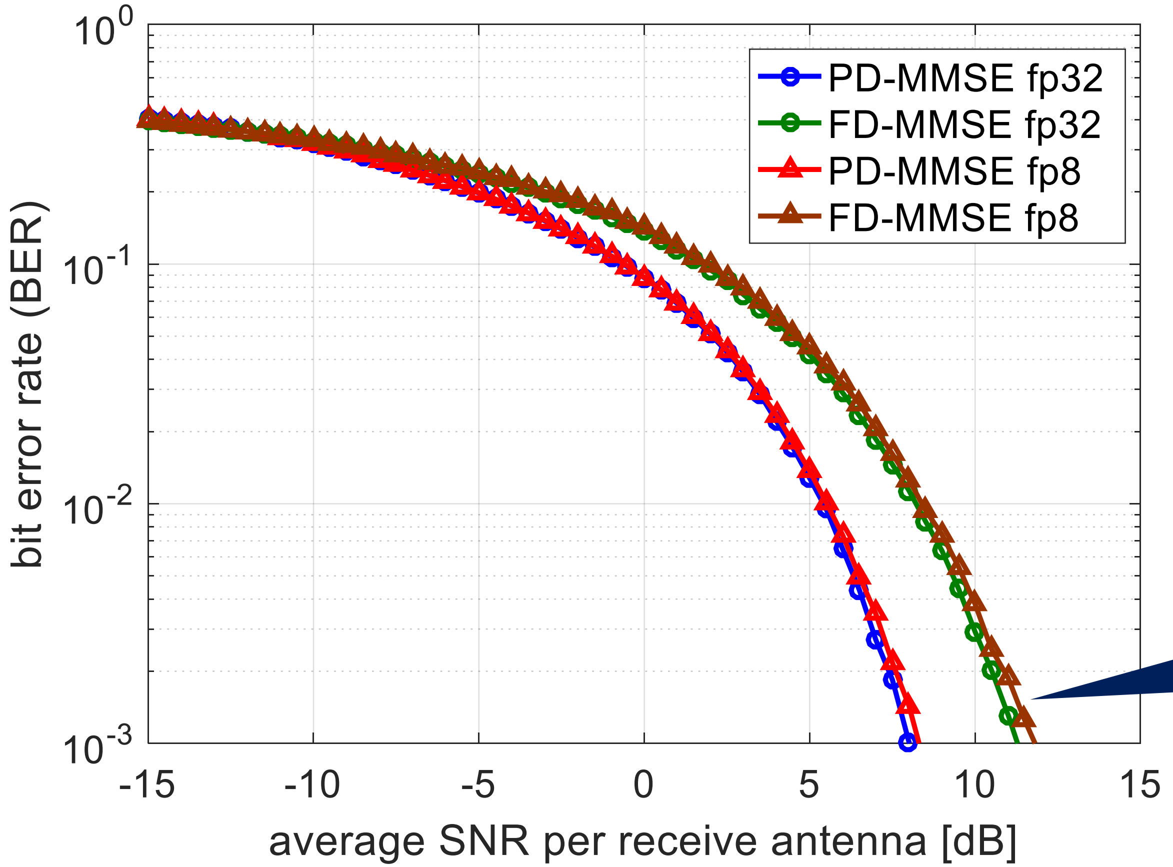


By integrating, ZF only requires 65% of the multiplies

Precision Trade-offs: 32-bit vs. 8bit floating point

Example: PD-MMSE and FD-MMSE

C=4, U=16, B_c=32, B=128, 16QAM, Quadriga NLOS urban campus channel

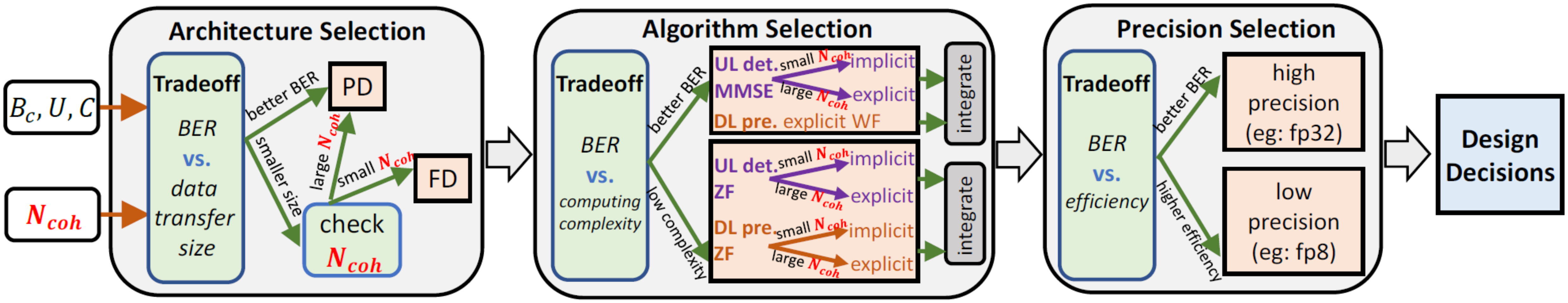


fp8: S | 4 expo | 3 man

FP8 is minor performance loss with significant savings

8-bit floating point reduces 4x data transfer size compared to 32-bit at only little loss of BER

Summary of Tradeoffs



Conclusion

- Decentralized baseband processing resolves complexity and interconnection bandwidth bottlenecks for massive MU-MIMO
- Practical massive MU-MIMO should leverage design trade-offs at different aspects:
  - Architecture trade-offs of PD and FD on BER vs. data transfer size
    - *Unless you expect very low coherence time, choose partially decentralized.*
  - Algorithms trade-offs of explicit and implicit methods on BER vs. complexity
    - *Use implicit matrix inversions whenever possible. Reuse results from uplink to downlink.*
  - Precision trade-offs of various data precision options on BER vs. efficiency
    - *Use fp16 or even fp8 unless BER is serious concern.*