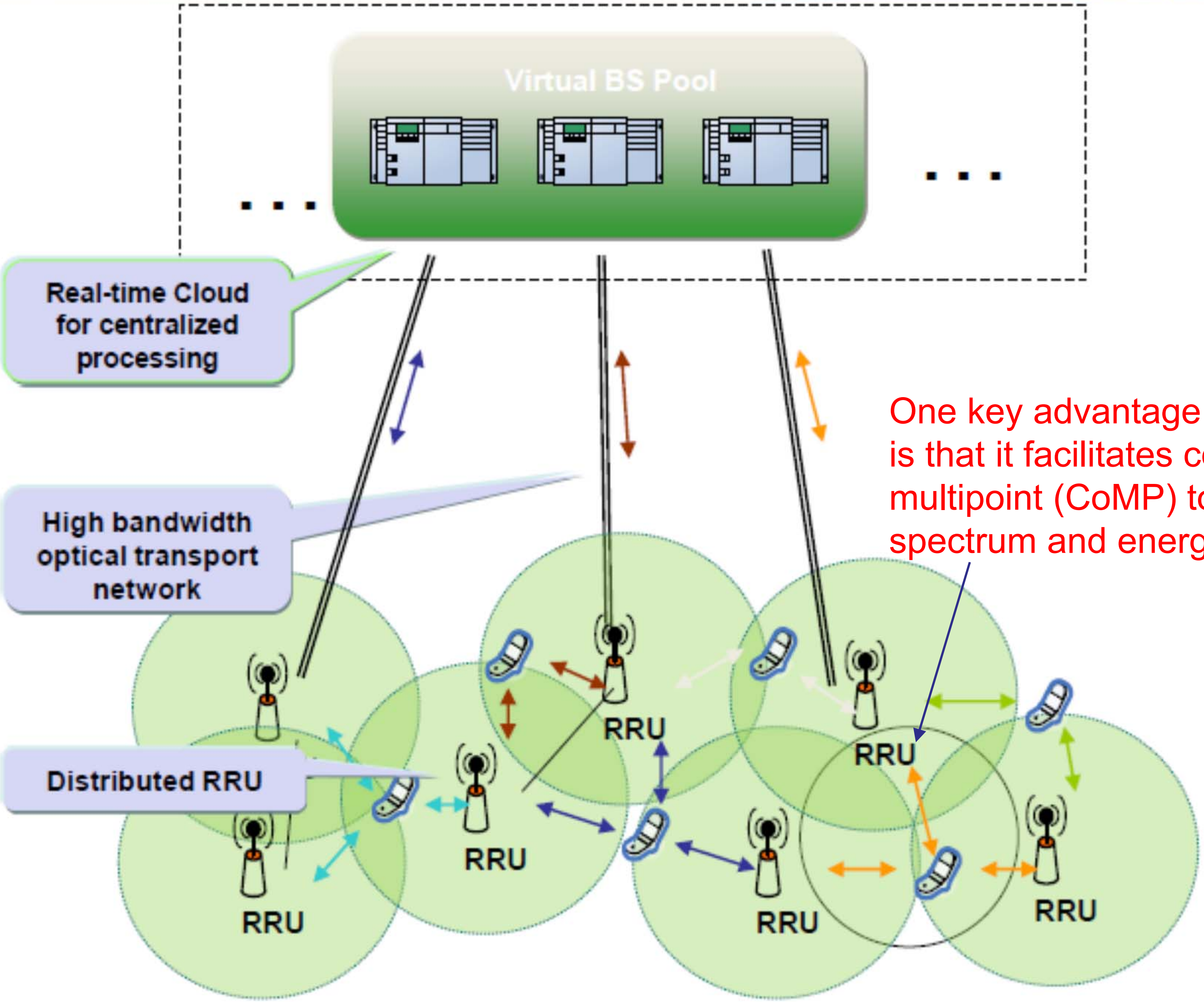
香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Randomized User-Centric Clustering for Cloud Radio Access Network with PHY Caching

An Liu, Vincent LAU and Wei Han
the Hong Kong University of Science and Technology

Background

Cloud Radio Access Networks (C-RAN)



One key advantage of C-RAN is that it facilitates coordinated multipoint (CoMP) to improve spectrum and energy efficiency

Issue 1: Joint Clustering and Interference Mitigation (Precoding)

- In C-RAN, as the number of RRHs increases, the overhead of full CoMP among all RRHs increases dramatically
  - The amount of feedback needed from the users increases
  - The CSI feedback and other processing delay increases which may cause further performance degradation
- Scalable Solution → clustering techniques are required
  - Network centric clustering (NCC)
    - The CRAN is divided into a set of non-overlapping RRH clusters (virtual BSs (VBSs))
    - The performance of the UEs at the boundary of the clusters will be compromised
  - User-centric clustering (UCC)
    - Each UE chooses a small number of RRHs as serving VBS
    - There can be overlap among VBSs to avoid cell edge effect. Hence UCC usually outperforms NCC.
    - However, User-centric clustering is challenging since the clusters are chosen in a dynamic way and may overlap
- One side effect of RRH clustering is the inter-cluster interferences among different VBSs
  - Efficient joint clustering and precoding is essential for practical deployment of C-RAN

Drawbacks of the Exiting Joint Clustering and Precoding Schemes

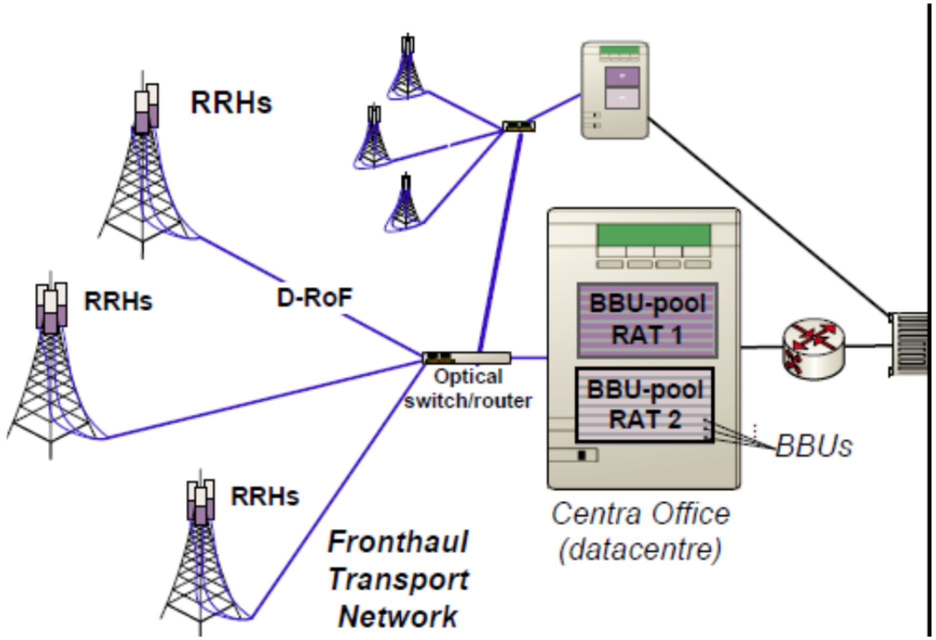
- One-timescale Centralized UCC (e.g., group sparse beamforming [1])
  - Requires real-time global CSIT -> huge CSI signaling overhead + sensitive to signaling latency
  - Large computation complexity
  - Not scalable to large network
- Heuristic two-timescale schemes (e.g., [2])
  - The RRH clustering is updated at slower timescale based on channel statistics
  - The precoder is updated at each time slot based on instantaneous CSI from the active RRHs to the users
  - Lower computational complexity and CSI signaling overhead
  - The RRH clustering and precoding solutions are obtained in a heuristic manner (i.e., the solution is not derived from a single joint optimization problem
  - The performance gap w.r.t. the optimal solution can be large.

[1] B. Dai and W. Yu, "Sparse beamforming and user-centric clustering for downlink cloud radio access network," IEEE Access, vol. 2, pp. 1326–1339, 2014.
[2] A. Liu and V. Lau, "Joint power and antenna selection optimization in large cloud radio access networks," IEEE Trans. Signal Processing, vol. 62, no. 5, pp. 1319–1328, March 2014.

Issue 2: Fronthaul Issues

- Fronthaul transport network (between BBUs and RRHs):
  - Digital Radio over Fibre (D-RoF).
  - Using typically the Common Public Radio Interface (CPRI) standard.
- Digitation requires high bit-rate CPRI links:
  - Site with 3 RRHs (LTE, 20MHz) requires 7.4 Gbit/s link.
  - Site with 15 RRHs (LTE-A (2 bands), 3G (2 bands), 2G (1 band)) requires up to 20Gbit/s link.

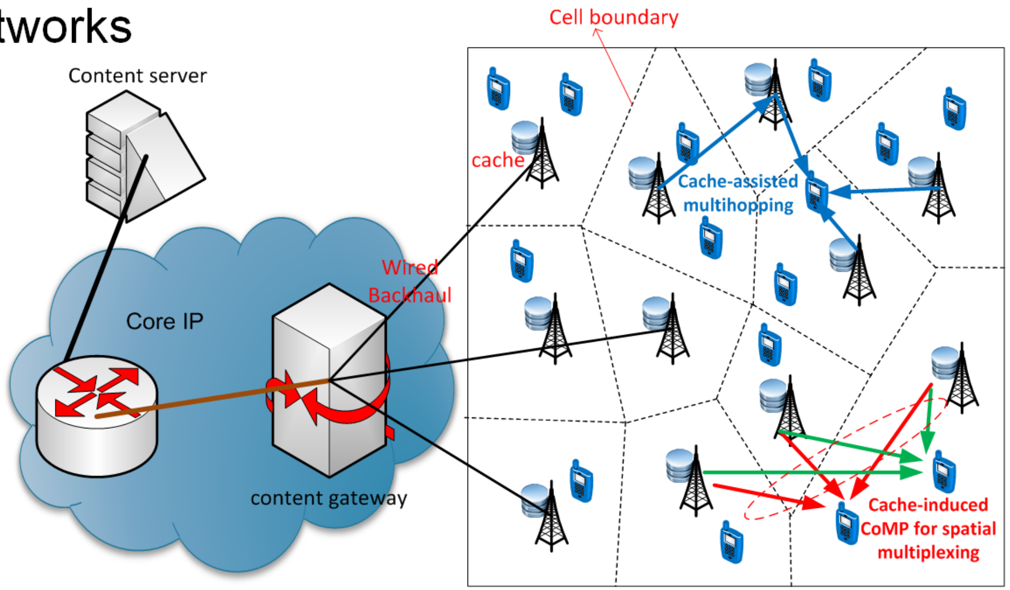
- Low latency: Maximum round trip delay of 150µs (~15km optical fiber).
- Jitter and synchronization:
  - Stringent requirements for frequency and phase synchronization.



Q: How to reduce the required fronthaul loading to reduce the cost and energy consumption of franthaul?

Can we exploit caching to reduce fronthaul loading?

- Motivation:
  - More than 50% of the wireless traffic comes from content delivery applications such as video streaming
  - Content are cachable at BS, e.g., the popular content cached at the BS is likely to be requested by many users later
  - Wireless caching: Cache the popular content at the BS during off-hours to improve the end-to-end performance and reduce backhaul/fronthaul loading at peak-hours
- PHY caching has been proposed to reduce backhaul cost and enhance capacity in dense wireless networks
  - ➢ Each BS has a cache
  - ➢ Only a small fraction of BSs have payload backhauls -> reduced backhaul cost
  - ➢ The role of backhaul is replaced by cheap cache without sacrificing order of capacity

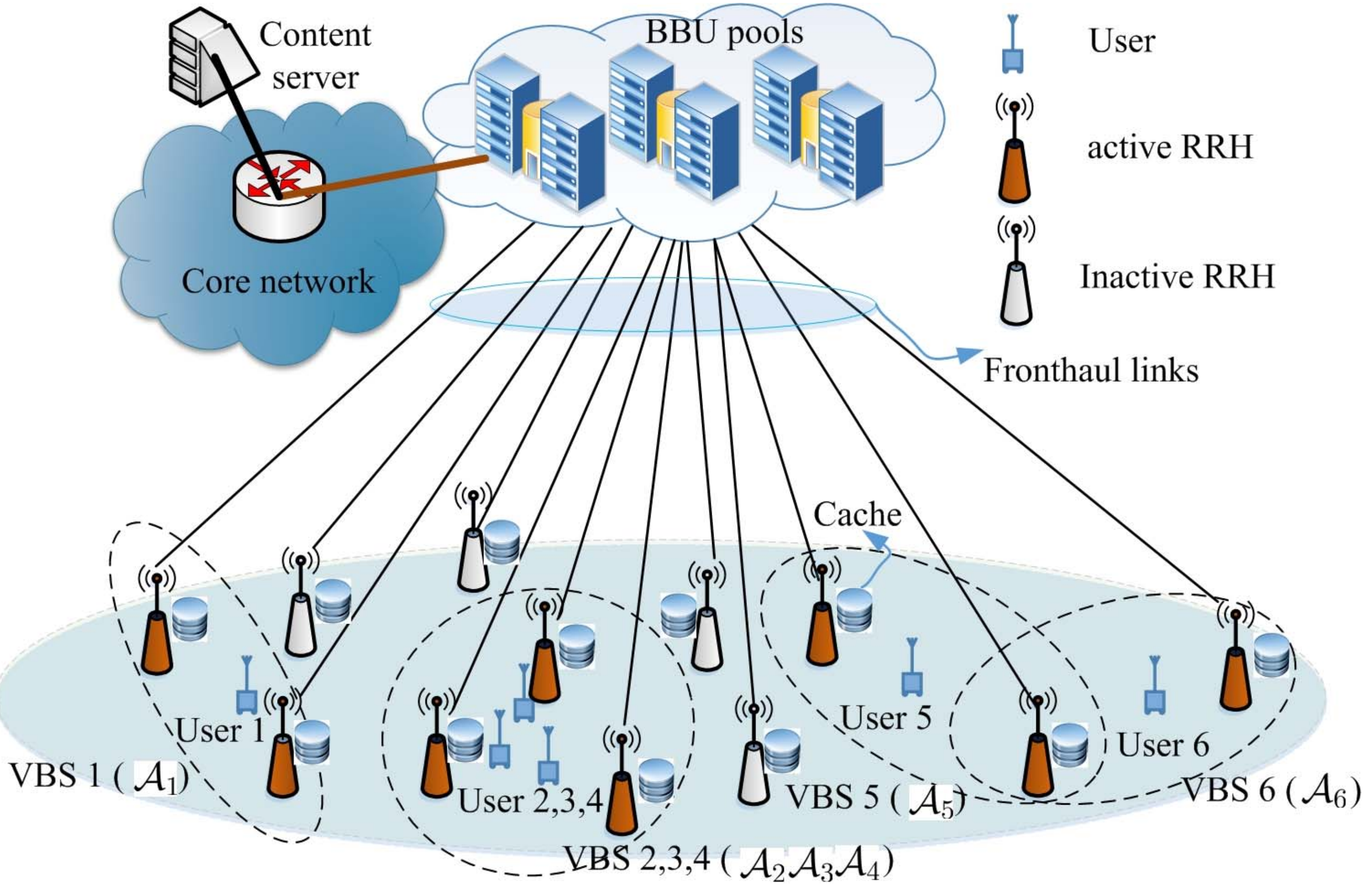


Q: Can we reduce the fronthaul loading by caching some popular content at the RRHs?

Mutli-timescale Optimization Framework & Challenges

- We propose a mutli-timescale optimization framework to address the above two issues of C-RAN, which contain the following key components
  - Radio Interference Processing
    - User-centric RRH Clustering: medium-term control adaptive to the channel statistics
    - Precoding: Short-term control adaptive to the instantaneous CSI from the active RRHs to the users at each time slot
  - PHY caching at RRH: long-term control adaptive to content popularity
    - PHY caching at the RRH is used to further reduce the fronthaul loading
- Challenges
  - Non-convex Stochastic Optimization of Precoding and RRH Clustering: In the mixed timescale optimization, the objective function (average WSR) involves the optimal short-term precoding solutions, which do not have closed form expressions. Moreover, the UCC RRH clustering belongs to combinatorial optimization
  - PHY Caching at RRH with Limited Processing Capability and Cache Capacity: In C-RAN where most baseband processing such as channel coding is implemented in the BBU, the RRH can no longer directly cache the original content packets.
  - Efficiency of PHY Caching in C-RAN: In addition to algorithm designs, it is very important to have a fundamental understanding of the tradeoff between the PHY cache capacity and the fronthaul loading in C-RAN as well as how such tradeoffs are affected by various key system parameters such as the total content size and content popularity distribution.

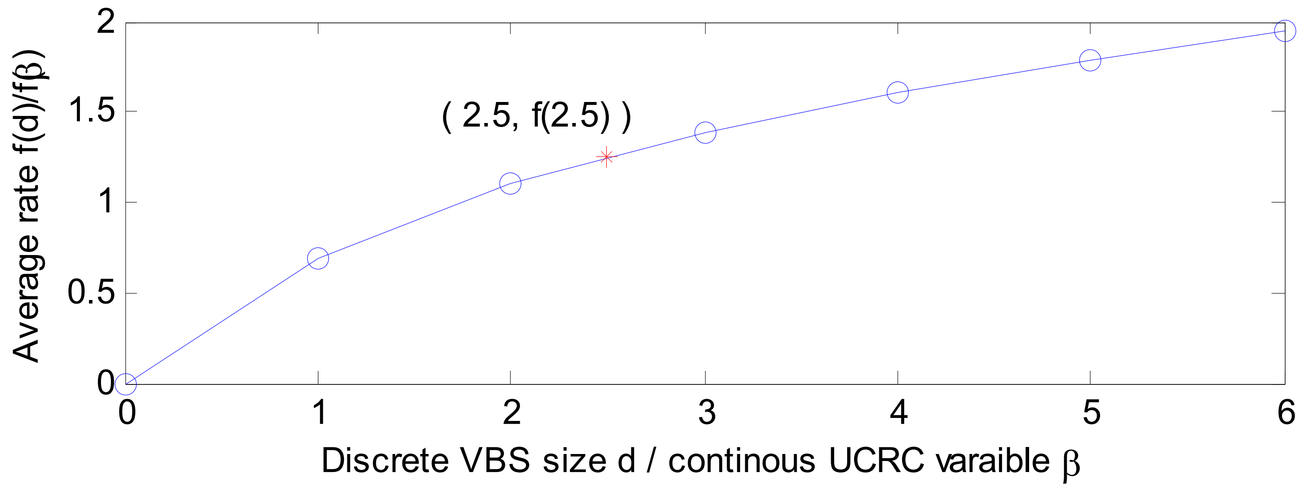
System Model



- A BBU connected with M RRHs serving K single-antenna users
- Each RRH has N_T antennas and a cache of size B_C
- Each user communicates with the nearest d_k RRHs, denoted by $\mathcal{A}_k(d_k)$ (VBS k)

Randomized User-centric RRH Clustering (UCRC)

- Let $\mathbf{d} \triangleq [d_1, ..., d_K]^T$ denote the VBS size vector
- The optimization of **d** belongs to discrete optimization, which is NP-hard
- Solution: propose a randomized UCRC with parameter $\boldsymbol{\beta} = [\beta_1, ..., \beta_K]^T$ (real vector) to make the problem continuous
- Toy example with K = 1



  - For given β , e.g., β = 2.5, the VBS size d is randomly generated from 2 candidate VBS sizes {2,3} according to the PMF [0.5,0.5]
  - In other words, if we observe the realizations of the VBS size d for 100 time slots, there are about 50 time slots with VBS size size d = 2 and 50 time slots with VBS size size d = 3
  - The average achievable rate is f(2.5) = 0.5*f(2) + 0.5*f(3)

Randomized User-centric RRH Clustering (UCRC)

- For general cases with arbitrary number of users K
  - For given $\boldsymbol{\beta}$, the VBS size d is randomly generated from K+ 1 candidate VBS size vectors $\{\mathbf{d}_1(\boldsymbol{\beta}), ..., \mathbf{d}_{K+1}(\boldsymbol{\beta})\}$ according to the PMF $\boldsymbol{\rho}(\boldsymbol{\beta}) = [\rho_1(\boldsymbol{\beta}), ..., \rho_{K+1}(\boldsymbol{\beta})]^T$
  - The candidate VBS size vectors and the PMF can be calculated using Algorithm 1
  - The candidate VBS size vectors is the coordinates of the K + 1 vertices of the simplex sub-region that contains $\boldsymbol{\beta}$, as illustrated in Figure 2.
  - The average achievable rate is $\hat{f}(\boldsymbol{\beta}) = \sum_{n=1}^{K+1} \rho_n(\boldsymbol{\beta}) f(\mathbf{d}_n(\boldsymbol{\beta}))$

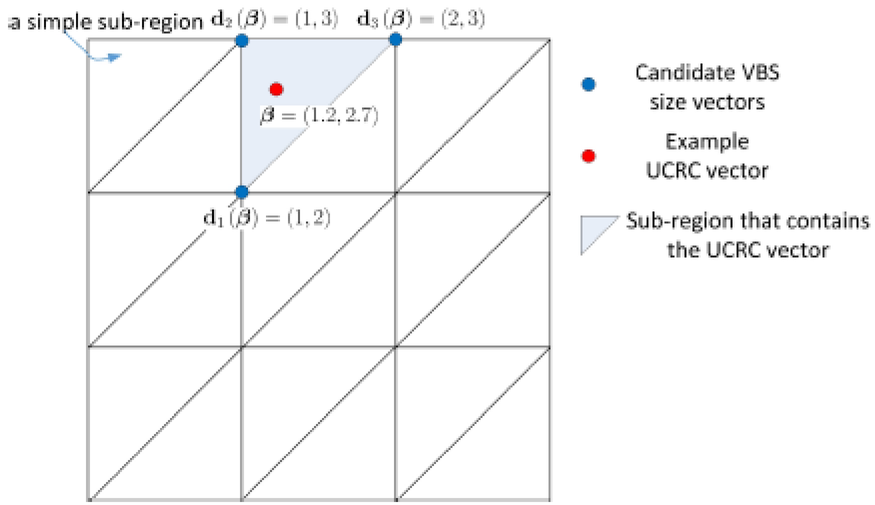


Figure 2: An illustration of randomized user-centric RRH clustering and the simplex sub-regions. Under UCRC vector $\boldsymbol{\beta} = [1.2, 2.7]^T$, the candidate VBS size vectors are $\left\{[1,2]^T, [1,3]^T, [2,3]^T\right\}$ and the PMF vector is $[0.3, 0.5, 0.2]^T$, i.e., at each time slot, the VBS size vector $\mathbf{d}$ is randomly generated from $\left\{[1,2]^T, [1,3]^T, [2,3]^T\right\}$ according to the PMF $[0.3, 0.5, 0.2]^T$.

Algorithm 1 Generate $\{\mathbf{d}_1(\boldsymbol{\beta}), ..., \mathbf{d}_{K+1}(\boldsymbol{\beta})\}$ and $\boldsymbol{\rho}(\boldsymbol{\beta})$

Step 1: Let $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} - \lfloor \boldsymbol{\beta} \rfloor$.
Step 2: Sort the component of $\tilde{\boldsymbol{\beta}}$ to obtain $\tilde{\beta}_{\kappa(\boldsymbol{\beta},1)} \geq \tilde{\beta}_{\kappa(\boldsymbol{\beta},2)} \geq ... \geq \tilde{\beta}_{\kappa(\boldsymbol{\beta},K)}$, where $\{\kappa(\boldsymbol{\beta},1), ..., \kappa(\boldsymbol{\beta},K)\}$ is a permutation of $\{1, ..., K\}$ and it depends on $\boldsymbol{\beta}$. Let $\mathbf{d}_1(\boldsymbol{\beta}) = \lfloor \boldsymbol{\beta} \rfloor$, $\mathbf{d}_n(\boldsymbol{\beta}) = \mathbf{d}_{n-1}(\boldsymbol{\beta}) + \mathbf{e}_{n-1}, n = 2, ..., K+1$, where $\mathbf{e}_{n-1}$ is the unit vector with the $\kappa(\boldsymbol{\beta}, n-1)$-th element equal to 1.
Step 3: Let $\rho_n(\boldsymbol{\beta}) = \tilde{\beta}_{\kappa(\boldsymbol{\beta},n-1)} - \tilde{\beta}_{\kappa(\boldsymbol{\beta},n)}, n = 1, ..., K+1$, where $\tilde{\beta}_{\kappa(\boldsymbol{\beta},0)} = 1$ and $\tilde{\beta}_{\kappa(\boldsymbol{\beta},K+1)} = 0$.

Simplex-interpolation algorithm

Problem Formulation for Radio Interference Processing

- Medium-term UCRC policy:

Channel statistics

Total VBS size constraint

$$\Omega_\beta = \{\beta(\Psi) \in \mathcal{D}_\beta : \forall \Psi\} \quad \mathcal{D}_\beta = \left\{\beta : \beta_k \in [0, M], \forall k \text{ and } \sum_{k=1}^{K} \beta_k \le \beta_T\right\}.$$

- Short-term precoding policy:

Instantaneous CSI from the active RRHs to the users

$$\Omega_v = \{\mathbf{V}(\mathbf{d}, \mathbf{h}_\mathcal{A}) \in \mathcal{D}_v(\mathbf{d}) : \forall (\mathbf{d} \in \mathcal{D}_d, \mathbf{h}_\mathcal{A})\}$$

$$\mathcal{D}_d = \{\mathbf{d} : d_k \in \mathbb{Z}_+, d_k \le M, \forall k\}.$$

$$\mathcal{D}_v(\mathbf{d}) = \left\{\mathbf{V} : \mathbf{v}_k \in \mathbb{C}^{N_T d_k}, \forall k \text{ and } \sum_{k=1}^{K}\sum_{j=1}^{d_k} \|\mathbf{v}_{k,j}\|^2 \mathbb{I}(m = \mathcal{A}_{k,j}) \le P, \forall m\right\}$$

Per RRH power constraint

- Joint UCRC and precoding optimization formulation

$$\mathcal{P} : U(\Omega_\beta, \Omega_v) \triangleq \max_{\Omega_\beta, \Omega_v} \mathbb{E}\left[\sum_{k=1}^{K} \mu_k \overline{r}_k(\beta(\Psi), \Omega_v)\right]$$

Conditional average rate for given UCRC and precoding policies

Problem Decomposition Radio Interference Processing

- Using primal decomposition, P is equivalent to the following families of subproblems

Timescale $\mathbb{T}_S$ subproblem (Short-term Precoding for given $(\mathbf{d} \in \mathcal{D}_d, \mathbf{h}_\mathcal{A})$):

$$\mathcal{P}_S(\mathbf{d}, \mathbf{h}_\mathcal{A}) = \max_{\mathbf{V} \in \mathcal{D}_v(\mathbf{d})} \sum_{k=1}^{K} \mu_k r_k(\mathbf{d}, \mathbf{V}, \mathbf{h}_\mathcal{A}).$$

Timescale $\mathbb{T}_M$ subproblem (Medium-term Clustering for given Ψ):

$$\mathcal{P}_M(\Psi) = \max_{\beta \in \mathcal{D}_\beta} \sum_{k=1}^{K} \mu_k \overline{r}_k(\beta, \Omega_v^\star),$$

where $\Omega_v^\star = \{\mathbf{V}^\star(\mathbf{d}, \mathbf{h}_\mathcal{A}) : \forall (\mathbf{d} \in \mathcal{D}_d, \mathbf{h}_\mathcal{A})\}$ is the optimal precoding policy

$\boxed{\text{optimal solution of } \mathcal{P}_S(\mathbf{d}, \mathbf{h}_\mathcal{A})}$

The optimal UCRC policy is given by: $\Omega_\beta^\star = \{\beta^\star(\Psi) : \forall \Psi\}$

$\boxed{\text{optimal solution of } \mathcal{P}_M(\Psi)}$

Outline of the Solution for Radio Interference Processing

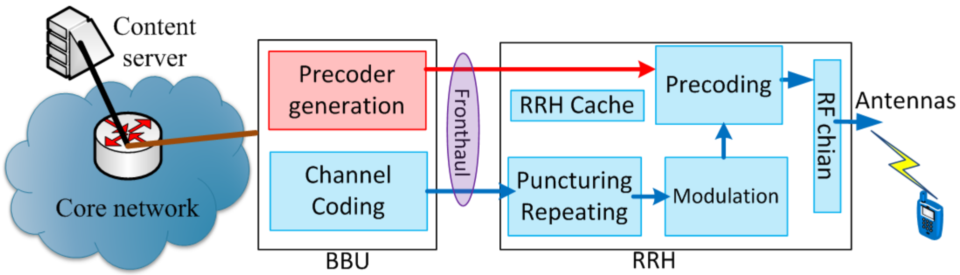
- $\mathcal{P}_S(\mathbf{d}, \mathbf{h}_{\mathcal{A}})$ can be solved using the WMMSE approach [3]
  - Since the problem is non-convex, WMMSE only finds a stationary point which may not be the global optimal solution
- For given stationary pre-coding policy $\Omega_v^* = \{\mathbf{V}^*(\mathbf{d}, \mathbf{h}_{\mathcal{A}}) : \forall (\mathbf{d} \in \mathcal{D}_d, \mathbf{h}_{\mathcal{A}})\}$, the objective (average WSR) of $\mathcal{P}_M(\boldsymbol{\Psi})$ is piece-wise linear

  $\boxed{\textit{stationary point of } \mathcal{P}_S(\mathbf{d}, \mathbf{h}_{\mathcal{A}})}$ found by WMMSE

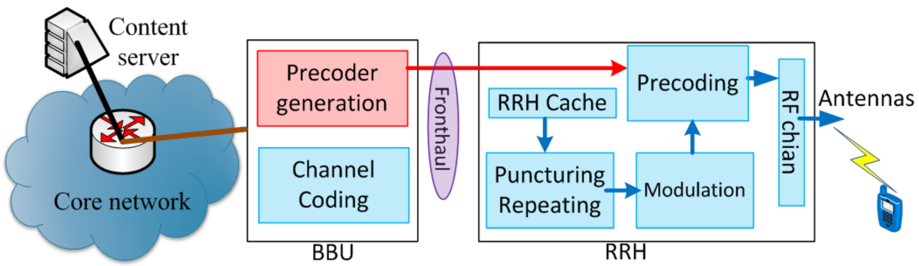
- We propose a gradient-projection-like (GP-like) algorithm to solve it
  - Again, the problem is non-convex and GP-like algorithm cannot ensure global convergence
  - Moreover, GP-like algorithm requires knowledge of channel statistics to calculate the average WSR for given UCRC vector
- Q: Can we design an online self-learning algorithm which ensures global convergence without explicit knowledge of channel statistics?
- The above question will be addressed in the journal version
  - A local stochastic cutting plane algorithm (SCPA) is proposed to solve $\mathcal{P}_M(\boldsymbol{\Psi})$
  - We can establish the global convergence of the local SCPA, i.e., the local SCPA converges to a solution whose gap from the global optimal solution can be bounded
  - In the week interference regime where the distance between users is large, it can be shown the local SCPA converges to the global optimal solution of $\mathcal{P}_M(\boldsymbol{\Psi})$

Systematic Channel Coded PHY Caching at RRH

- Consider content delivery application
  - there are L files on the content server, the size of the l-th file is F_l bits
  - each user independently accesses the l-th file with probability p_l
- Caching at RRH can be used to reduce the fronthaul loading
  - Since RRH can only perform simple baseband processing, the RRH cannot cache the original content files
- We propose a systematic channel coded PHY caching scheme
  - Each file is divided into packets of size N_S bits at the content server
  - For each packet, a systematic channel codeword with length N_S/c and coding rate c is generated
  - During off-peak hours, each RRH caches the systematic channel codewords of randomly chosen $p_l^* F_l/N_S$ packets of the l-th file for all l
  - In the online payload transmission phase, if the systematic channel codeword of the packet requested by user k is in the cache of the serving RRHs, the serving RRHs directly transmit the systematic channel codeword (after puncturing or repeating) to user k



(a) If the codeword of the packet requested by user k is not in the RRH cache, the serving RRHs will obtain it from the BBU via franthaul and transmit it to user k



(b) Otherwise, the serving RRHs directly obtains the corresponding codeword from the local caches without consuming the fronthaul and transmit it to user k

Problem Formulation for PHY Caching

- The cache content placement vector $\mathbf{q} = [q_1, \ldots q_L]^T$ determines the relative priority of caching the L files
  - It must be carefully chosen to minimize the fronthaul loading under the cache capacity constraint at each RRH
- For given UCRC and precoding policy at PHY, the total average fronthaul loading is

$$R_F(\mathbf{q}) = R_S \left(1 - \sum_{l=1}^{L} p_l q_l\right)$$

Average sum rate under given UCRC and precoding policy

- The cache optimization is formulated as a fronthaul loading minimization:

Long-term Caching Problem: $\mathcal{P}_L \ : \min_{\mathbf{q} \in \mathcal{D}_q} 1 - \sum_{l=1}^{L} p_l q_l$

cache capacity constraint

$$\mathcal{D}_q = \left\{ \mathbf{q} : \ q_l \in [0,1], \forall l \text{ and } \sum_{l=1}^{L} q_l F_l \leq c B_C \right\}$$

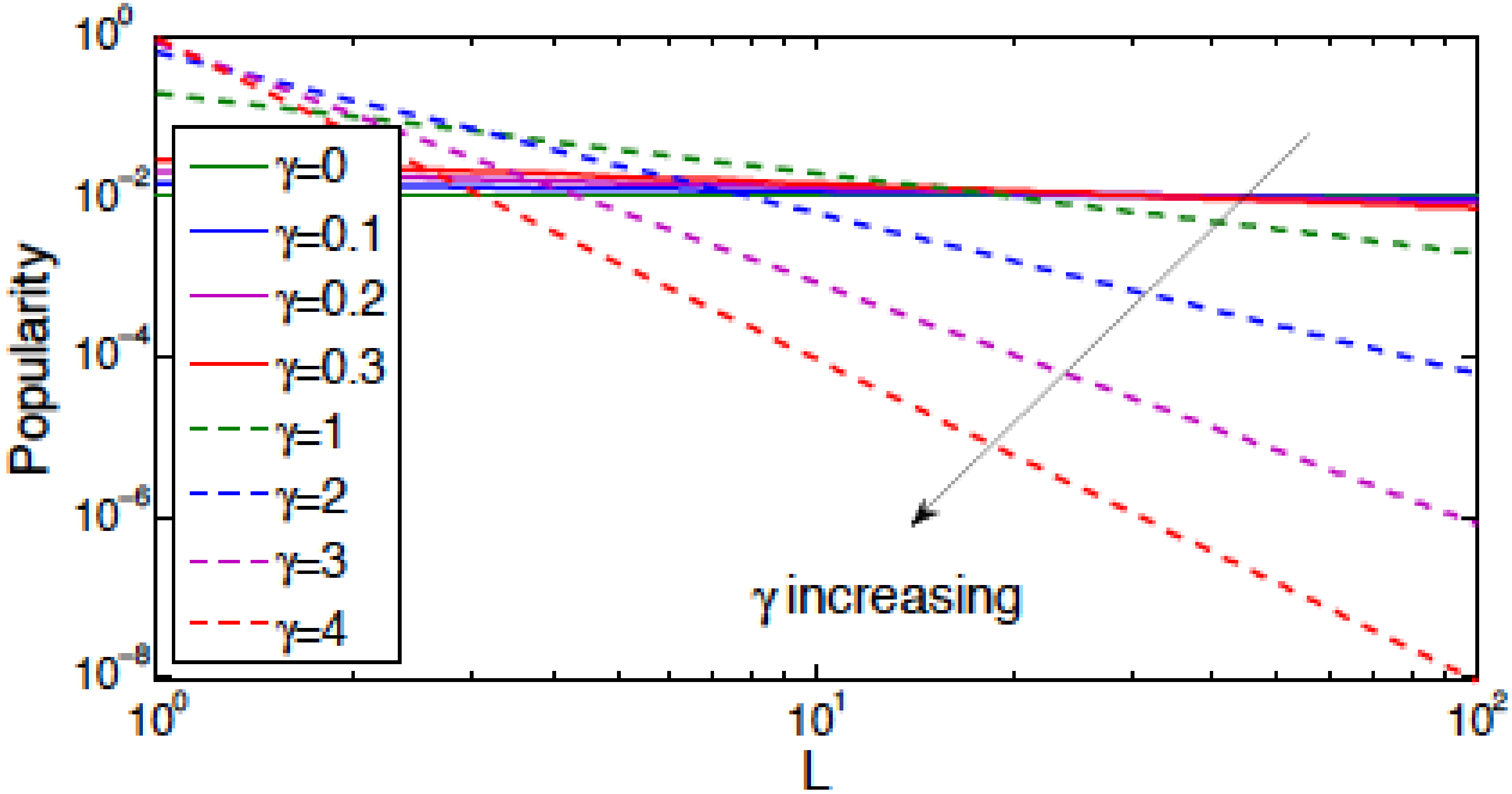
Solution for PHY Caching

- If the popularity distribution p is known, the optimal solution of the LP problem P_L can be easily obtained using numerical method.
- If p is unknown, P_L is a stochastic LP problem and the optimal solution can be solved using a stochastic subgradient algorithm
- We focus on analyzing the tradeoff between cache capacity B_C and the fronthaul loading $R_F(q)$ under the Zipf popularity distribution

Popularity skewness

Normalization factor $$p_l = \frac{1}{Z_\gamma(L)} l^{-\gamma}, l = 1, 2, \ldots, L$$

- The Zip distribution is widely used to model the Internet traffic
  - On the Internet, Zipf's law appears to be the rule rather than the exception
  - A larger popularity skewness τ implies that the user requests concentrate more on a few popular files
  - Large popularity skewness τ is usually observed in wireless applications



(a) Zipf popularity under large γ on a log-log scale. The user requests concentrate more on a few content files as γ is increasing.

Cache-fronthaul Tradeoff Analysis under Zipf Popularity Distribution

- Under Zipf Law, the optimal cache content placement vector is to cache the most popular files, i.e.,

$$q_l^\star = \begin{cases} 1, & l \le \lfloor b_C \rfloor \\ b_C - \lfloor b_C \rfloor, & l = \lfloor b_C \rfloor + 1 \\ 0, & \text{otherwise} \end{cases}$$

$$b_C = \frac{cB_C}{F}$$

normalized cache capacity

- The minimum fronthaul loading for given normalized cache capacity is

$$R_F^\star = \left(1 - \frac{\sum_{l=1}^{\lfloor b_C \rfloor} l^{-\gamma} + (b_C - \lfloor b_C \rfloor)(\lfloor b_C \rfloor + 1)^{-\gamma}}{Z_\gamma(L)}\right) R_S$$

- Define the *fronthaul gain* over the case without caching as

$$\triangle r_F^\star = \frac{R_s - R_F^\star}{R_s} = \frac{\sum_{l=1}^{\lfloor b_C \rfloor} l^{-\gamma} + (b_C - \lfloor b_C \rfloor)(\lfloor b_C \rfloor + 1)^{-\gamma}}{Z_\gamma(L)}$$

The fraction of the reduced fronthaul loading due to PHY caching

Cache-fronthaul Tradeoff Analysis under Zipf Popularity Distribution

- Impact of key system parameters on cache-fronthaul tradeoff
  - **Impact of the normalized cache capacity** b_C**:** As the normalized cache capacity b_C increases from 0 to L, the fronthaul loading decreases to 0, and the fronthaul gain increases from 0 to 1.
  - **Impact of the number of content files L:** The fronthaul loading increases as L increases. When L= Θ (1) , the fronthaul gain is Θ (1). When L → ∞, the fronthaul gain depends heavily on the popularity skewness γ .
  - **The impact of the popularity skewness** γ is summarized in the following Theorem.

Theorem 5 (Asymptotic fronthaul gain for large L). *When $L \to \infty$ and $b_C = \Theta(1)$, the asymptotic scaling laws of $\triangle r_F^\star$ are summarized below:*

- **Sub-critical:** If $\gamma < 1$, then $\triangle r_F^\star = \Theta\left(L^{-(1-\gamma)}\right)$.
- **Critical:** If $\gamma = 1$, then $\triangle r_F^\star = \Theta\left(1/\ln(L)\right)$.
- **Super-critical:** If $\gamma > 1$, then $\triangle r_F^\star = \Theta(1)$.

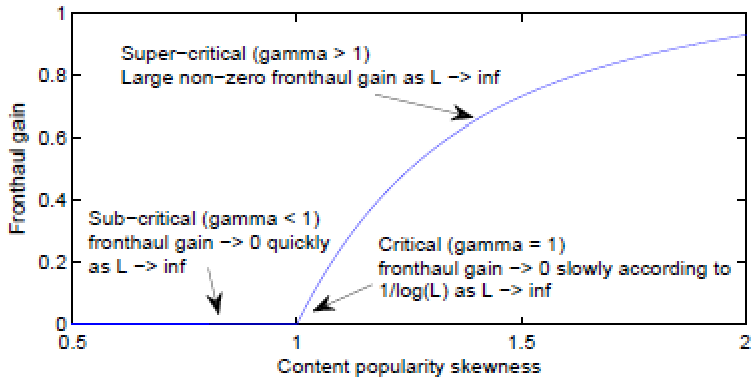
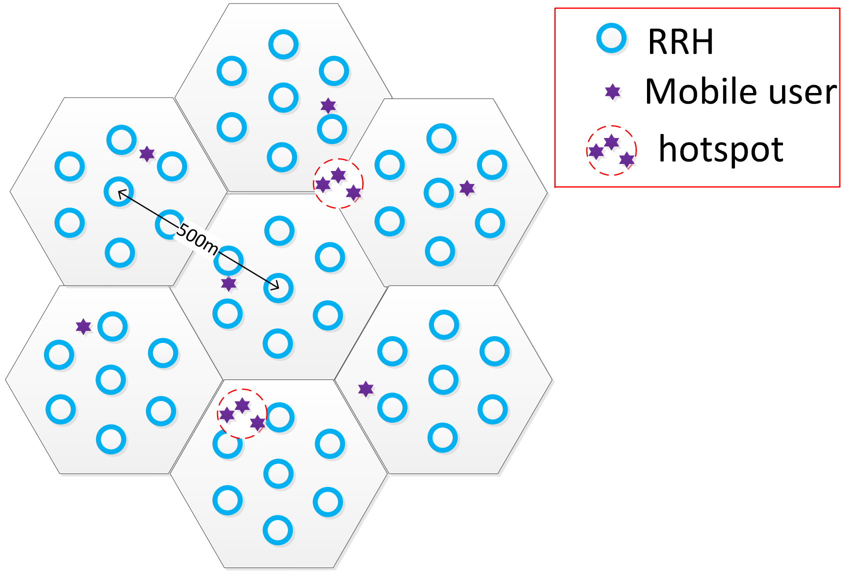


Figure 4: Phase transition behavior of asymptotic fronthaul gain for large L with $b_C = 10$.

In practice, the popularity skewness γ can be large, especially for mobile applications. In this case, it is still possible to achieve a large fronthaul gain, even when the cache capacity B_C is relatively small compared to the total content size L*F .

Simulation Configuration



- A C-RAN with 49 RRHs
- 7 regular hexagon cells
- 12 randomly distributed users
- Each RRH has two antennas
- There are L=50 content files
- The size of each file is 1GB
- Zipf popularity distribution with γ =1.5
- **Baseline 1 - One-timescale GSBF** in [1] with total fronthaul loading constraint
- **Baseline 2 - Static RRH Clustering**: A fixed number of the nearest d RRHs are chosen to serve each user. The short-term precoding is the same as the proposed scheme.
- **Baseline 3 -** Proposed user-centric RRH clustering without caching.

Simulation Results

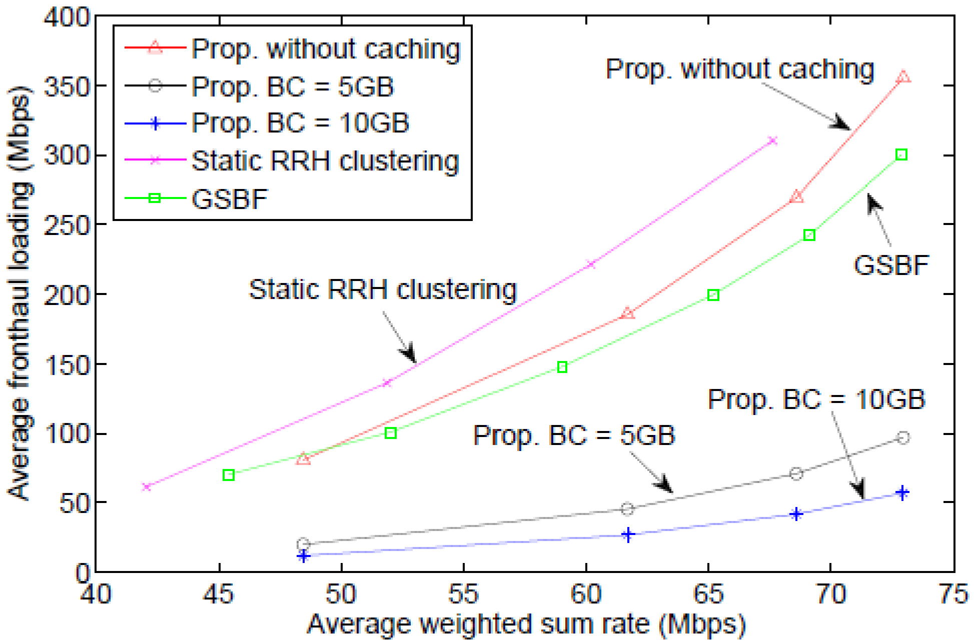


Figure 9: Average fronthaul loading versus average WSR.

- Proposed scheme without caching
  - achieve a tradeoff performance close to the one-timescale GSBF
  - better than the static RRH clustering
- Proposed With RRH-level caching
  - significantly outperforms the one-timescale GSBF
  - performance gain increases as the cache capacity increases
  - less CSI signaling overhead and lower computational complexity than the one-timescale GSBF

Conclusion

- We propose a mutli-timescale optimization framework to optimize the tradeoff performance in C-RAN
  - The mixed-timescale radio interference processing is formulated as a joint UCRC and precoding optimization problem
    - A WMMSE-based algorithm to find a stationary point for the short-term precoding
    - a self-learning local SCPA (in the journal version) to solve the medium-term UCRC subproblem with a provable performance bound.
    - The proposed solution is asymptotically optimal for the joint problem in the weak interference regime
  - The long-timescale PHY caching is formulated as a fronthaul loading minimization problem
    - The optimal cache replacement can be obtained by solving a LP
    - There is a phase transition behavior in the cache-franthaul tradeoff
      - As the total content size goes to infinity with fixed cache capacity, a large caching gain is still achievable when the popularity skewness is larger than one
      - but the caching gain is zero otherwise