Cross-Layer Resource Allocation With Elastic Service Scaling in Cloud Radio Access Network

Md.Al-Helal
Roll:SH-51

Jobayed Ullah
Roll:EK-107

Computer Science & Engineering
CSEDU

February 25, 2018

Cross-Layer Resource Allocation With Elastic Service Scaling in Cloud Radio Access Network

Jianhua Tang
Wee Pen Tay
Tony Q. S. Quek

IEEE Transactions on Wireless Communications, vol 14, no. 9

September 2015

Assumption

Assumption

$$\mu_i, c_i > \lambda_i$$

$$\varphi_i(\mu_i) \geq 0, \forall \mu_i$$

$\varphi_i(\mu_i)$ is a convex and increasing function of μ_i

- μ_i service rate
- c_i transmission rate
- λ_i arrival rate
- $\varphi_i(\mu_i) = k_i \mu_i^{a_i}$
- $k_i > 0$ and $a_i > 1$ are positive constants.

Aim

Minimizing system power consumtion in C-RAN, which consists of three components:

- Power consumption in BBU pool
- Power consumption in fiber links
- Power consumption in RRHs

System Model

Delay

$$d_i = \frac{1}{\mu_i - \lambda_i} + \frac{1}{c_i - \lambda_i}$$

where,

- d_i total delay
- μ_i service rate
- λ_i arival rate
- c_i transmission rate

System Model

Recieved signal at UE i

$$\hat{x}_i = \sum_{j \in \mathcal{A}} h_{ij}^H w_{ij} x_i + \sum_{k \neq i}^{N} \sum_{j \in A} h_{ij}^H w_{kj} x_k + \delta_i$$

where,

- $\mathcal{A}$ set of active RRHs
- x_i data symbol for ith user
- $w_{ij} \in \mathbb{C}^k$ transmit beamformer for UE i from RRH j
- $h_{ij}^H \in \mathbb{C}^k$ channel from RRH j to UE i
- $\delta_i \sim \mathcal{CN}(0, \sigma_i)$ additive white Gaussian noise(AWGN) at UE i
- $i \in \mathcal{N}$
- $j \in \mathcal{A}$

System Model

Signal-to-Interference-plus-Noise-Ratio(SINR) at UE i

$$\mathsf{SINR}_i(\mathcal{A}) = \frac{\left| \sum_{j\in\mathcal{A}} h_{ij}^H w_{ij} \right|^2}{\sigma_i^2 + \sum_{k\neq i}^{N} \left| \sum_{j\in\mathcal{A}} h_{ij}^H w_{kj} \right|}$$

where,

- σ_i noise
- $\mathcal{A}$ set of active RRHs
- $w_{ij} \in \mathbb{C}^k$ transmit beamformer for UE i from RRH j
- $h_{ij}^H \in \mathbb{C}^k$ channel from RRH j to UE i
- $i \in \mathcal{N}$
- $j \in \mathcal{A}$

System Model

The achieveable rate c_i of UE i should satisfy

$$c_i \leq B_i \log(1 + \mathsf{SINR}_i(\mathcal{A}))$$

where,

- B_i bandwidth for UE i
- SINR Signal-to-Interference-plus-Noise-Ratio
- $\mathcal{A}$ set of active RRHs

Transmitting Power

Each RRH j has maximum transmitting power constraint

$$\sum_{i=1}^{N} w_{ij}^{H} w_{ij} = \sum_{i=1}^{N} \|w_{ij}\|^2 \leq E_j$$

where,

- $w_{ij} \in \mathbb{C}^k$ transmit beamformer for UE i from RRH j
- $i \in \mathcal{N}$
- $j \in \mathcal{L}$

Problem Formulation

$$\min_{\mu_i, c_i, w_{ij}, \mathcal{A}} \sum_{i}^{N} \varphi_i(\mu_i) + |\mathcal{A}| P_f + \frac{1}{\eta} \sum_{i=1}^{N} \sum_{j \in A} w_{ij}^H w_{ij}$$

subject to

$$\frac{1}{\mu_i - \lambda_i} + \frac{1}{c_i - \lambda_i} \leq \tau_i$$

$$\lambda_i < \mu_i, \lambda_i < c_i$$

$$c_i i \leq B_i \log(1 + \mathsf{SINR}_i(\mathcal{A}))$$

$$\sum_{i=1}^{N} w_{ij}^H w_{ij} \leq E_i, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{L}$$

- $\varphi_i(\mu_i)$ VM i's power consumption
- μ_i VM i's service rate
- P_f power consumption of active fibre links
- $\eta \in (0, 1)$ inefffficiency coefficient of amplifier in RRH
- $\mathcal{A}$ set of active RRHs
- $w_{ij} \in \mathbb{C}^k$ transmit beamformer for UE i from RRH j
- $i \in \mathcal{N}$
- $j \in \mathcal{A}$

Quasi Weighted Sum-Rate Maximization Problem

Quasi Weighted Sum-Rate Maximization Problem

QWSRM

$$\min_{c_i, w_{ij}} \sum_{i=1}^{N} -\varepsilon_i c_i$$

subject to,

$$c_i \le B_i \log(1 + \mathsf{SINR}_i(\mathcal{A})), \quad \forall i \in \mathcal{N}$$

$$\sum_{i=1}^{N} w_{ij}^H w_{ij} \le E_i, \quad \forall j \in \mathcal{A}$$

- ε_i — arbitrary nonnegative weight
- c_i — transmission rate
- B_i — bandwidth for UE i
- $\mathcal{A}$ — set of active RRHs
- E_j — maximum power on a link
- $w_{ij} \in \mathbb{C}^k$ — transmit beamformer for UE i from RRH j

Generalization of WSRM

$$\min_{c_i, w_{ij}} f(\mathbf{c})$$

subject to,

$$\sum_{i=1}^{N} w_{ij}^{H} w_{ij} \leq E_i, \quad \forall j \in \mathcal{A}$$

Where the objective function $f(\mathbf{c})$, for $0 \leq \mathbf{c} \leq \bar{\mathbf{c}}$ has the following properties:

- $f(\mathbf{c})$ is a function only of $\mathbf{c}$, and
- $f(\mathbf{c}) < \infty$ is continuously differentiable, and
- $f(\mathbf{c})$ is convex in the feasible region
- $w_{ij} \in \mathbb{C}^k$ transmit beamformer for UE i from RRH j
- c_i transmission rate
- E_j maximum power on a link
- $\mathcal{A}$ set of active RRHs
- $\mathbf{c} = [c_1, \ldots, c_N]^T$

$$\gamma_{ub}(\mathcal{Q}) = \begin{cases} f(\mathbf{c}_{min}), & \mathbf{c}_{min} \in \mathcal{F} \\ +\infty, & otherwise \end{cases}$$

$$\gamma_{lb}(\mathcal{Q}) = \begin{cases} f(\mathbf{c}_{max}), & \mathbf{c}_{max} \in \mathcal{F} \\ +\infty, & otherwise \end{cases}$$

where,

- $\mathbf{c}_{min} = \left[c_{1,min}, \ldots, c_{N,min}\right]^T$
- $\mathbf{c}_{max} = \left[c_{1,max}, \ldots, c_{N,max}\right]^T$
- $\gamma_{ub}(\mathcal{Q})$ upper bound
- $\gamma_{lb}(\mathcal{Q})$ lower bound

BnB algorithm for QWSRM problem

Input: $\mathcal{Q}_{\text{init}}, \mathcal{A}$, and$\{$(c).

Initialize: Obtain c_i by solving $\frac{\delta f(c)}{\delta c_i} = 0$, for $i \in \mathcal{N}$. Set $k = 1, \mathcal{B} = \mathcal{Q}_{init}, u_1 = \gamma_{ub}(\mathcal{Q}_{init})$ and $l_1 = \gamma_{lb}(\mathcal{Q}_{init})$.

Check the feasibility of problem (17) with given

if feasible **then**

$c_0 = \widetilde{c}$;

else

while $u_k - l_k > \epsilon$ **do**

Branching:

- Set $\mathcal{Q}_k = \mathcal{Q}$, where $\mathcal{Q}$ satisfies $\gamma_{lb}(\mathcal{Q}) = l_k$.
- Split $\mathcal{Q}$ into $\mathcal{Q}_{\text{I}}$ and $\mathcal{Q}_{\text{II}}$, along one of its longest edges.
- Update $\mathcal{B}_{k+1} = (\mathcal{B}_k \setminus \{\mathcal{Q}_k\}) \bigcup (\mathcal{Q}_{\text{I}}, \mathcal{Q}_{\text{II}})$.

Bounding:

- Update $u_{k+1} = \min_{\mathcal{Q} \in \mathcal{B}_{k+1}} \{\gamma_{ub}(\mathcal{Q})\}$
- Update $l_{k+1} = \min_{\mathcal{Q} \in \mathcal{B}_{k+1}} \{\gamma_{lb}(\mathcal{Q})\}$

end while

Set $c_0 = c_{min}$;

end if

Output: c_0.

Thank You