CoMP: JT

October 1, 2023

1 Simulation Strategies

1.1 Notations Used

Before introducing our simulation strategies, we shall introduce some of the notations we have maintained throughout our experiments.

Notation	**Details**
r	Radius of each BS in the network
N	No. of UEs per BS
M	Total no. of BSs in the network
RB_{max}	Maximum limiting no. of resource-blocks available per BS
λ_{step}	Increment/Decrement of step-size of chi, value is $1/RB_{max}$
λ_j	Traffic/chi for BS_j
$\Re_j$	Ring-ID for BS_j
RB_j	No. of resource blocks **used** for BS_j
$Pr_{i,j}$	Recieved power (mW) of UE_i from BS_j
$\gamma_{i,j}$	SINR of UE_i from BS_j
$T_{i,j}$	Throughput received by UE_i from BS_j
$d_{i,j}$	Distance (m) of UE_i from BS_j
$\varrho_{i,j}$	Is the UE_i of BS_j dropped ? [0, 1]
$\kappa_{i,j}$	Is the UE_i for BS_j considered for calculation of metrics? [0, 1]
$\tau_{i,j}$	Effective throughput for UE_i under BS_j

Table 1: Notations and their details used throughout the simulations.

1.2 Pseudocodes

In this section, we shall introduce the simulation strategy algorithms used for various simulations conducted.

1.2.1 Basic Simulation Block

Algorithm 1: Simulation procedure basic block, per chi(per monte-carlo) for a particular no. of co-ordinating BSs.

Input: Chi Value: λ, No. of co-ordinating BSs: ϖ, Radial distance to put UEs: d_r (Optional), Tier: *tier* (Optional)

Output: Throughput array (T) of each UE per BS i.e.$\forall_i \forall_j T_{i,j}$

$\sum_{j=1}^{M} RB_j := 0$ `// Initialize each resource-blocks used to 0`

if *tier not specified* **then**

  Use tier = 3 (for dummy ring calculations - see later), and place 37 base stations wrt hexagonal structure.

else

  Use tier = *tier*, and place base $1 + 3tier(tier + 1)$ base stations wrt hexagonal structure.

for $j \leftarrow 1$ *to* M `// Iterate through BSs`

do

  if *Radial distance is not specified* **then**

    Place $\forall_k$ UE$_k$ for this BS randomly within radius r

  else

    Place $\forall_k$ UE$_k$ for this BS at random angular distance, but constant radial distance of d_r

  `// UE placements and initial calculations`

  for $i \leftarrow 1$ *to* N `// Iterate through UEs`

  do

    $d_{i,j} \leftarrow$ Distance of UE$_i$ from BS$_j$

    $fade_{i,j} \leftarrow \mathcal{N}(\mu = 0, \sigma = 8)$ `// Fading effect`

    $Pr_{i,j} \leftarrow Pt - [FsPL + 10 * \alpha * \frac{d_{i,j}}{d_0} + fade_{i,j}]$ `// Convert to mW`

    Sort BSs wrt received powers $Pr_{i,j}$ in descending order

    if $\forall_k \; BS_k \in [BS_1, BS_2, ..., BS_{\varpi}] \; \exists_k \; s.t. \; RB_k == RB_{max}$: **then**

      $\varrho_{i,j} \leftarrow 1$ `// All co-ordinating BSs don't have available RBs`

    if $\varrho_{i,j} == 0$: `// Accept the UE` **then**

      $RB_j \leftarrow RB_j + 1$

      $\gamma_{i,j} \leftarrow \frac{\sum_{k \in co-ordinating} Pr_{i,k}}{P_{noise} + \sum_{k \in competing} Pr_{i,k}}$ `// SINR calculation`

      $T_{i,j} \leftarrow 180 * log_2(1 + \gamma_{i,j})$ `// Throughput calculation`

  `// Calculation after UE placements`

  $\lambda_j \leftarrow \frac{RB_j}{RB_{max}}$ `// traffic update for each BS`

  for $i \leftarrow 1$ *to* N `// Iterate through UEs`

  do

    $\gamma_{i,j} \leftarrow \frac{\sum_{k \in co-ordinating} Pr_{i,k}}{P_{noise} + \sum_{k \in competing} \lambda_k * Pr_{i,k}}$ `// SINR after adjusting traffic`

    $T_{i,j} \leftarrow 180 * log_2(1 + \gamma_{i,j})$ `// Throughput calculation`

return array T

We use this basic strategy block of simulation procedure for futher simulations.

1.2.2 Avg metrics vs chi

We utilize the basic simulation block as described in 1 to compute various metrics.

Algorithm 2: Simulation for varying traffic and varying co-ordinating BSs using monte-carlo of 1000.

Initialize T array for receiving throughputs per 1000 monte-carlo steps.
for $\lambda := 1; \lambda >= 0; \lambda := \lambda - \lambda_{step}$ **do**
  for $bs_{coord} := 0; bs_{coord} <= 5; bs_{coord} := bs_{coord} + 1$ **do**
    for $mc \leftarrow 1$ *to* 1000 **do**
      $T_{i,j,bs_{coord},mc} \leftarrow$ basic-simulation-block(λ, bs_{coord})
Compute the average of $T_{i,j,\lambda}$, for each bs_{coord}, for 1000 monte-carlo steps.

1.2.3 Hourly traffic variation

We utilize the basic simulation block as described in 1 to compute various metrics for the hourly traffic variation.

Algorithm 3: Simulation for hourly traffic variation.

Initialize T array for receiving throughputs per 1000 monte-carlo steps.
for $\lambda \in \lambda_{hourly}$ **do**
  for $bs_{coord} := 0; bs_{coord} <= 5; bs_{coord} := bs_{coord} + 1$ **do**
    for $mc \leftarrow 1$ *to* 1000 **do**
      $T_{i,j,bs_{coord},mc} \leftarrow$ basic-simulation-block(λ, bs_{coord})
Compute the average of $T_{i,j,\lambda}$, for each bs_{coord}, for 1000 monte-carlo steps.

1.2.4 Metrics variation vs distance for particular traffic values

We utilize the basic simulation block as described in 1 to compute various metrics for distance variation with respect to the appointed Base-Station for particular values of traffic.

Algorithm 4: Distance based variation.

Initialize T array for receiving throughputs per 1000 monte-carlo steps.
for $\lambda \in [0.1, 0.2, ..., 1.0]$ **do**
  for $bs_{coord} := 0;\ bs_{coord} <= 5;\ bs_{coord} := bs_{coord} + 1$ **do**
    for $r := 0.1;\ r <= 1000;\ r := r + 100$ **do**
      for $mc \leftarrow 1$ *to* 1000 **do**
        // Vary radial distance
        $T_{i,j,bs_{coord},mc} \leftarrow$ basic-simulation-block(λ, bs_{coord}, r)
Compute the average of $T_{i,j,\lambda,r}$, for each bs_{coord}, for 1000 monte-carlo steps.

1.2.5 Tier-wise variation of metrics

We utilize the basic simulation block as described in 1 to compute various metrics for distance variation with respect to the appointed Base-Station for particular values of traffic.

Algorithm 5: Variation in terms of tier (dummy vs no dummy).

Initialize T array for receiving throughputs per 1000 monte-carlo steps.
for $tier \leftarrow 1$ *to* 12 **do**
  for $bs_{coord} := 0;\ bs_{coord} <= 5;\ bs_{coord} := bs_{coord} + 1$ **do**
    for $mc \leftarrow 1$ *to* 1000 **do**
      // Use chi = 1, vary tier
      $T_{i,j,bs_{coord},mc} \leftarrow$ basic-simulation-block$(1, bs_{coord}, tier)$
Compute the average of $T_{i,j,tier}$, for each bs_{coord}, for 1000 monte-carlo steps.

1.2.6 Metrics calculations

In this section we calculate the metrics that we have computed throughout our simulations per condition eg. per chi, per no. of co-ordinating BSs, per tier. To compute metrics, we are given the throughput array, $\forall_i \forall_j T_{i,j}$ and the drop-UE array, $\forall_i \forall_j \varrho_{i,j}$.

For dummy ring, we use the array κ i.e. $\forall_i \forall_j \kappa_{i,j}$, where

$$\kappa_{i,j} = \begin{cases} 1 & \text{, if } \Re_j < tier \\ 0 & \text{, else} \end{cases}$$

We use the effective throughput array, τ i.e. $\forall_i \forall_j \tau_{i,j}$ to compute the rest of the metrics, where $\tau_{i,j} = T_{i,j} * \varrho_{i,j} * \kappa_{i,j}$

Average throughput is calculated by the formula in 1.

$$T_{avg} = \frac{1}{NM} * \sum_{j=1}^{M} \sum_{i=1}^{N} \tau_{i,j} \tag{1}$$

Similar to average throughput in 1, various other metrics are calculated. Spectral efficiency is computed by the formula in 2

$$S_{avg} = \frac{1}{B} * \sum_{j=1}^{M} \sum_{i=1}^{N} \tau_{i,j} \tag{2}$$

Jain's Fairness Index is computed by the formula in 3

$$F = \frac{1}{NM} * \frac{(\sum_{j=1}^{M} \sum_{i=1}^{N} \tau_{i,j})^2}{\sum_{j=1}^{M} \sum_{i=1}^{N} \tau_{i,j}^2} \tag{3}$$

We have also computed proportion of UE dropped (as well as active) for a particular chi according to 4 and 5

$$p_{dropped} = \frac{1}{NM} * \sum_{j=1}^{M} \sum_{i=1}^{N} [\varrho_{i,j} == 1] \tag{4}$$

$$p_{active} = \frac{1}{NM} * \sum_{j=1}^{M} \sum_{i=1}^{N} [\varrho_{i,j} == 0] \tag{5}$$

Finally, we have computed effective chi for a particular chi using the formula $\lambda_{eff} = p_{active} * \lambda$. Ideally, λ_{eff} and λ should have the same value, however according to our simulations, and considering the dropped user-end-devices, λ_{eff} plateaus after a particular value of effective-traffic, owing to the intuition that no matter how many co-ordinating base stations we increase, after a certain traffic input, not all the traffic will be served, and some will be dropped.