List of Abbreviations

BLER Block Error Rate
MCS Modulation and Coding Scheme
MRT Maximum Ratio Trasmission
OLLA Outer Loop Link Adaptation
PRB Physical Resource Block
TB Transport Block
ZF Zero Forcing

1

Implementation

Contents

1.1 System-level Simulator

This chapter is not finished. It should be read along with the notes in PowerPoint.

Main Steps in the System-level Simulatior:

1. Application traffic...
2. variable initialisation
3. SLS loop: Update CSI, Update scheduled grants, Simulate TTI

Steps on 'Update CSI', if the previous TTI was a CSI TTI:

1. Update the best GoB precoders for each UE, for each layer.
2. Update the interference measurements for each UE, for each layer.

Steps on 'Update scheduling', if the previous TTI was a Scheduling TTI:

1. Which UEs to consider for scheduling: the ones with something to send
2. Select BS per UE (Place holder): The BS with the best beam for a given UE is considered as the serving BS.
3. Select SU-MIMO setting on the number of layers: one layer or two layers. Choose the one that gives highest aggregated bitrate. One layer transmits with double the power per layers (since it uses the power assigned to the other layer as well) and the selected layer is the one that has the highest SINR. gives better SINR since the reception can be separate and the received powers summed together, but two layers have double the data streams. The reception is done with both polarisations in both cases.

   To estimate the bitrate, we do the following:

   (a) Estimate the MCS from the MCS tables: pick the first CQI that gets less than 0.1 BER (or BLER?)

   (b) (Optional) Adjusts the MCS choice with the OLLA parameter. It can be UE specific or global.

   (c) From the MCS and the assigned resources, estimate the achievable bitrate.
4. Compute priorities based on the estimated instantaneous bitrate. Depending on the scheduler, it may consider the latencies as well.
5. Select MU-MIMO setting: list the users that will be co-scheduled. The co-scheduling rule is to add the layer or layers of a UE to the list, by order of UE priority, if the best beams used for those layers are compatible with the previously added UE layers. We define as imcompatible layers with beam-pairs from the GoB that are at less than k beams apart. If k is 0, then the all layers are accepted. If $k = 1$, then the beams must be different - adjacent beams have a distance of 1. Beams located diagonally adjacent of the GoB are considered to have a distance of 2, hence they may be co-schedule with $k = 1$ and $k = 2$. Beam distance is defined by the sum of differences of the beam indices in the

grid. Also, incompatibility is only checked with layers from different users as the SU-MIMO choice before should have decided one the same user layers.

6. Perform power control: i) distribute the available power per uniformly per layer, respecting the maximum available power per antenna element, and how many layers that antenna element is going to transmit; ii) scale down precoders with non-uniform amplitudes on the weights and that are causing power overflows in certain antennas;
7. Re-estimate the SINRs, per UE, per layer, based on:
   (a) the number of layers each user will actually have, if it has changed from the last estimation
   (b) the newly assigned power
8. Choose the final MCS, per UE, per layer, based on the last SINR estimate

There are three noteworthy remarks about the above:

- It can be performed on a narrowband level: we may do the above for portions of the band. The only thing that needs to change across the the schedulable bands is the aggregated-across-time throughput metric in the scheduler, or else the resources would be attributed equality anyway. Therefore, instead of the actual aggregated-across-time throughtput, we should use and estimated aggregated-across-time throughtput, and update that estimation with the expected bitrate each user would get from the schedulable narrowband. For the first portion of band, the estimation and the actual values would be the same. Note that if the scheduler takes into account latencies as well, then the head-of-queue delay value provided to the scheduler must change across narrowbands, it should assume the bits are correctly sent and compute the latency of 'the next packet on the queue'. Or something even smarter, perhaps like considering a few more packets ahead of time, since it will take a while until the scheduling changes.
- It can be performed for UL as well, with small changes:
  - the PRBs attributed should depend on the channel quality of the UE, and a Power Density should be computed to achieve a certain SIRN per PRB
  - the interference needs to be computed differently for the uplink. Same principle, but different variables since other UEs are causing the interference.
  - the noise needs to be computed differently since we won't know how big the bandwidth will be ahead of time. For the first estimation, it is enough to consider the same bandwidth for all users, or to estimate based on the quality of the beam pair (channel gain measured when CSI measure was obtained). Then, when the actual priorities are defined, a new and more precise value can be used for the noise.
- It can be performed for Implicit beamforming, with small changes:
  - The first SINR estimation is performed with Maximum Ratio Trasmission (MRT) precoders;

- The second SINR estimation is performed with Zero Forcing (ZF) precoders - that computation requires the MU-MIMO setting;
- Anything else?!

- When more BSs are used, and when more beams besides the best are reported, something more intelligent can be made in the BS selection. Nonetheless, it is assumed that all BSs communicate at a very fine time scale between them, to choose how to best serve the UEs.

Steps on Simulating the TTI:

1. Get the Transport Block (TB) size from the Modulation and Coding Scheme (MCS) used and the allocated number of PRBs. There's a TB per bandwidth part.
2. Compute the realised SINR, for each scheduled UE:
   (a) Compute Intra-cell interference
   (b) Use default value of Inter-cell interference.
   (c) Compute Noise.
   (d) Compute RSS.
3. Compute Block Error Rate (BLER).
4. Flip a BLER-biased coin to determine if the block was well received or had errors
5. (In case Outer Loop Link Adaptation (OLLA) is used) Update the OLLA parameter based on success or not from the block transmission.
6. Update the time-aggregated throughput per UE

Bibliography