cuMAC API Reference

Table of contents

cuMAC API Data Structures

CumacCellGrpPrms

API data structure containing cell group information of the coordinated cells.

Field	Type	Description
nUe	uint16_t	Total number of selected UEs in a TTI of all coordinated cells. Value: 0 -> 65535
nActiveUe	uint16_t	Total number of active UEs of all coordinated cells. Value: 0 -> 65535
numUeSchdPerCellTTI	uint8_t	Number of UEs selected/scheduled per TTI per cell. Value: 0 -> 255
nCell	uint16_t	Total number of coordinated cells. Value: 0 -> 65535
nPrbGrp	uint16_t	Total number of PRGs per cell. Value: 0 -> 65535
nBsAnt	uint8_t	Number of BS antenna ports. Value: 0 -> 255
nUeAnt	uint8_t	Number of UE antenna ports. Value: 0 -> 255
W	float	Frequency bandwidth (Hz) of a PRG. Value: 12 * subcarrier spacing * number of PRBs per PRG
sigmaSqrd	float	Noise variance. Value: noise variance value in watts
precodingScheme	uint8_t	Precoder type. Value: 0: No precoding 1: SVD precoder
receiverScheme	uint8_t	Receiver/equalizer type. Value: Currently only support 1: MMSE-IRC receiver
allocType	uint8_t	PRG allocation type. Value: 0: non-consecutive type-0 allocation 1: consecutive type-1 allocation
betaCoeff	float	Coefficient for adjusting the cell-edge UEs' performance in multi-cell scheduling Value: non-negative real number. The default value is 1.0, representing the classic proportional-fairness scheduling.

sinValThr	float	Singular value threshold for layer selection. Value: in (0, 1). Default value is 0.1
prioWeightStep	uint16_t	For priority-weight based scheduling algorithm. Step size for UE priority weight increment per TTI if UE does not get scheduled. Value: default 100
cellId	uint16_t[nCell]	IDs of coordinated cells. One dimensional array. Value of each element: Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. cellId[cIdx] is the ID of the cIdx-th coordinated cell
cellAssoc	uint8_t[nCell*nUe]	Cell-UE association indication for all the selected UEs of the coordinated cells. One dimensional array. Value of each element: Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. cellAssoc[cIdx*nUe + uIdx] == 1 means the uIdx-th selected UE is associated with cIdx-th coordinated cell, 0 otherwise.
cellAssocActUe	uint8_t[nCell*nActiveUe]	Cell-UE association indication for all active UEs of the coordinated cells. One dimensional array. Value of each element: Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. Denote uIdx = 0, 1, ..., nActiveUe-1 as the global active UE index in the coordinated cells. cellAssocActUe[cIdx*nUe + uIdx] == 1 means the uIdx-th active UE is associated with cIdx-th coordinated cell, 0 otherwise.
prgMsk	uint8_t[nCell][nPrbGrp]	Per-cell bit map for the availability of each PRG for allocation Two-dimensional array. Value of each element: Denote cIdx = 0, 1, ... nCell-1 as the coordinated cell index Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index prgMsk[cIdx][prgIdx] is the availability indicator for the prgIdx-th PRG in the cIdx-th coordinated cell 0 - unavailable, 1 - available
postEqSinr	float[nActiveUe*nPrbGrp*nUeAnt]	Array of the per-PRG per-layer post-equalizer SINRs of all active UEs in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global active UE index in the coordinated cells. Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote layerIdx = 0, 1, ..., nUeAnt-1 as the layer index.

		postEqSinr[uIdx*nPrbGrp* nUeAnt + prgIdx*nUeAnt + layerIdx] is the uIdx-th active UE's post-equalizer SINR on the prgIdx-th PRG and the layerIdx-th layer.
wbSinr	float[nActiveUe *nUeAnt]	Array of wideband per-layer post-equalizer SINRs of all active UEs in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global active UE index in the coordinated cells. Denote layerIdx = 0, 1, ..., nUeAnt-1 as the layer index. wbSinr[uIdx*nUeAnt + layerIdx] is the uIdx-th active UE's wideband post-equalizer SINR on the layerIdx-th layer.
FP32: estH_fr or FP16: estH_fr_ half	FP32: cuComplex[nC ell][nUe* nPrbGrp*nBsA nt*nUeAnt] Or FP16: __nv_bfloat162[nCell][nUe* nPrbGrp*nBsA nt*nUeAnt]	Per-cell array of the narrow-band SRS channel estimate coefficients for the selected UEs in the coordinated cells. Two-dimensional array: the 1st dimension is for cells, and the 2nd dimension is for UEs, PRGs, and antenna ports. Value of each element: Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote bsAntIdx = 0, 1, ..., nBsAnt-1 as the BS antenna port index Denote ueAntIdx = 0, 1, ..., nUeAnt as the UE antenna port index estH_fr[cIdx] [uIdx* nPrbGrp*nBsAnt*nUeAnt + prgIdx* nBsAnt*nUeAnt + bsAntIdx*nUeAnt + ueAntIdx] is the complex channel coefficient between the cIdx-th cell and the uIdx-th selected UE in the cell group on the prgIdx-th PRG, the bsAntIdx-th BS antenna port and the ueAntIdx-th UE antenna port. (The above applies to the FP16 version as well)
prdMat	cuComplex[nU e*nPrbGrp* nBsAnt*nBsAn t]	Array of the precoder/beamforming weights for the selected UEs in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote inPortIdx = 0, 1, ..., nBsAnt-1 as the precoder input port index Denote outPortIdx = 0, 1, ..., nBsAnt-1 as the precoder output port index prdMat[uIdx*nPrbGrp* nBsAnt*nBsAnt + prgIdx* nBsAnt*nBsAnt + inPortIdx *nBsAnt + outPortIdx] is the precoder/beamforming weight of the uIdx-th selected UE in the coordinated cells on the prgIdx-th PRG and between the inPortIdx-th input port and the outPortIdx-th output port.

detMat	cuComplex[nUe*nPrbGrp*nUeAnt*nUeAnt]	Array of the detector/beamforming weights for the selected UEs in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote inPortIdx = 0, 1, ..., nUeAnt-1 as the detector input port index. Denote outPortIdx = 0, 1, ..., nUeAnt-1 as the detector output port index. detMat[uIdx*nPrbGrp*nUeAnt*nUeAnt + prgIdx*nUeAnt*nUeAnt + inPortIdx*nUeAnt + outPortIdx] is the detector weight of the uIdx-th selected UE on the prgIdx-th PRG and between the inPortIdx-th input port and the outPortIdx-th output port.
sinVal	float[nUe*nPrbGrp*nUeAnt]	Array of the per-UE, per-PRG, per-layer singular values obtained from SVD. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the 0-based UE index for the selected UEs in the coordinated cells. Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote layerIdx = 0, 1, ..., nUeAnt-1 as the layer index. sinVal[uIdx*nPrbGrp*nUeAnt + prgIdx*nUeAnt + layerIdx] is the UE uIdx's layerIdx-th largest singular value on PRG prgIdx For each UE and on each PRG, the singular values are stored in descending order.

cumacCellGrpUeStatus

API data structure containing the per-UE information of the coordinated cell group.

Field	Type	Description
avgRates	float[nUe]	Array of the long-term average data rates of the selected UEs in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. avgRates[uIdx] is the long-term average throughput of the uIdx-th selected UE in the coordinated cells.
avgRatesActUe	float[nActiveUe]	Array of the long-term average data rates of all active UEs in the

		coordinated cells One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global active UE index in the coordinated cells. avgRatesActUe[uIdx] is the long-term average throughput of the uIdx-th active UE in the coordinated cells.
prioWeightActUe	u int16_t[nActiveUe]	For priority-based UE selection. Priority weights of all active UEs in the coordinated cells One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global UE index for all active UEs in the coordinated cells. prioWeightActUe[uIdx] is the uIdx-th active UE's priority weight. 0xFFFF indicates an invalid element.
tbErrLast	int8_t[nUe]	Array of the selected UEs' transport block (TB) decoding error indicators of the last transmissions One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. tbErrLast[uIdx] is the uIdx-th selected UE's TB decoding error indicator of the last transmission. -1 - the last transmission is not a new transmission (is a re-transmission) 0 - decoded correctly 1 - decoding error ** Note that if the last transmission of a UE is not a new transmission, tbErrLast of that UE should be set to -1.
tbErrLastActUe	int8_t[nActiveUe]	TB decoding error indicators of all active UEs in the coordinated cells. One dimensional array. Value of

		each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global UE index for all active UEs in the coordinated cells. tbErrLastActUe[uIdx] is the uIdx-th active UE's TB decoding error indicator: -1 - the last transmission is not a new transmission (is a re-transmission) 0 - decoded correctly 1 - decoding error ** Note that if the last transmission of a UE is not a new transmission, tbErrLastActUe of that UE should be set to -1.
newDataActUe	int8_t[nActiveUe]	Indicators of initial transmission/retransmission for all active UEs. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nActiveUe-1 as the global UE index for all active UEs in the coordinated cells. newDataActUe[uIdx] is the indicator of initial transmission/retransmission for the uIdx-th active UE in the coordinated cells 0 – retransmission 1 - new data/initial transmission -1 indicates an invalid element
allocSolLastTx	For type-0 PRG allocation: int16 _t[nCell*nPrbGrp] For type-1 PRG allocation: int16_t[2*nUe]	The PRG allocation solution of the last transmission of the selected/scheduled UEs in the coordinated cells Format referring to the description for allocSol in the cumacSchdSol structure
mcsSelSolLastTx	int16_t[nUe]	MCS selection solution of the last transmission of the selected/scheduled UEs in the coordinated cells. Format referring to the description for mcsSelSol in the cumacSchdSol structure

layerSelSolLastTx	uint8_t[nUe]	Layer selection solution of the last transmission of the selected/scheduled UEs in the coordinated cells. Format referring to the description for layerSelSol in the cumacSchdSol structure

cumacSchdSol

API data structure containing the scheduling solutions.

Field	Type	Description
setSchdUePerCellTTI	uint16_t[nCell* numUeSchdPerCellTTI]	Set of global IDs of the selected UEs per cell per TTI. One dimensional array. Value of each element: Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. Denote i = 0, 1, ..., numUeSchdPerCellTTI-1 as the i-th selected UE in a given cell. setSchdUePerCel lTTI[cIdx*numUeSchdPerCellTTI + i] is within {0, 1, ..., nActiveUe-1} and represents the global active UE index of the i-th selected UE in the cIdx-th coordinated cell.
allocSol	For type-0 PRG allocation: int1 6_t[nCell*nPrbGrp] For type-1 PRG allocation: int16_t[2*nUe]	PRB group allocation solution for the selected UEs per TTI in the coordinated cells One dimensional array. Value of each element: For type-0 PRG allocation: Denote prgIdx = 0, 1, ..., nPrbGrp-1 as the PRG index. Denote cIdx = 0, 1, ..., nCell-1 as the coordinated cell index. allocSol[prgIdx*nCell + cIdx] indicates the selected UE index (0, 1, ..., nUe-1) that the prgIdx-th PRG is allocated to in the cIdx-th coordinated cell. -1 indicates that a given PRG in a cell is not allocated

		to any UE. For type-1 PRG allocation: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. allocSol[2*uIdx] is the starting PRG index of the uIdx-th selected UE. allocSol[2*uIdx + 1] is the ending PRG index of the uIdx-th selected UE plus one. -1 indicates that a given UE is not being allocated to any PRG.
pfMetricArr	float[array_size] array_size = nCell * the minimum power of 2 that is no less than nPrbGrp*n umUeSchdPerCellTTI	Array to store the computed PF metrics per UE and per PRG. Only used for type-1 PRG allocation. One dimensional array. GPU memory allocated for CUDA kernel execution. Not used externally. Memory should be allocated when initializing the cuMAC API. Value of each element: floating-type value of a computed PF metric.
pfIdArr	uint16_t [array_size] array_size = nCell * the minimum power of 2 that is no less than nPrbGrp*n umUeSchdPerCellTTI	Array to indicate the PRG and UE indices of the sorted PF metrics. Only used for type-1 PRG allocation. One dimensional array. GPU memory allocated for CUDA kernel execution. Not used externally. Memory should be allocated when initializing the cuMAC API. Value of each element: 0 -> 65535
mcsSelSol	int16_t[nUe]	MCS selection solution for the selected UEs per TTI in the coordinated cells One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. mcsSelSol[uIdx] indicates the MCS level for the uIdx-th selected UE in the coordinated

		cells. Range of each element: 0, 1, ..., 27 (Currently only support Table 5.1.3.1-2: MCS index table 2, 3GPP TS 38.214). -1 indicates an element is invalid.
layerSelSol	uint8_t[nUe]	Layer selection solution for the selected UEs per TTI in the coordinated cells. One dimensional array. Value of each element: Denote uIdx = 0, 1, ..., nUe-1 as the selected UE index in the coordinated cells. layerSelSol[uIdx] indicates the number of layers selected for the uIdx-th selected UE in the coordinated cells. Range of each element: 0, 1, ..., nUeAnt-1 The selected layers have singular values descending from the largest one.

cuMAC Scheduler Module API

Multi-cell proportional-fairness UE down-selection

Wrapper class and public member functions:

> class cumac::multiCellUeSelection public: // constructor multiCellUeSelection(); // destructor ~multiCellUeSelection(); // setup() function for per-TTI algorithm execution void setup(cumac::cumacCellGrpUeStatus* cellGrpUeStatus, cumac::cumacSchdSol* schdSol, cumac::cumacCellGrpPrms* cellGrpPrms, uint8_t in_enableHarq, cudaStream_t strm); // requires external synchronization // set in_enableHarq to 1 if HARQ is enabled; 0 otherwise // run() function for per-TTI algorithm execution void run(cudaStream_t strm); // requires external synchronization // parameter/data buffer logging function for debugging purpose void debugLog(); // for debugging only, printing out dynamic descriptor parameters

Multi-cell proportional-fairness PRB scheduler

Wrapper class and public member functions:

class cumac::multiCellScheduler public: // constructor multiCellScheduler(); // destructor ~multiCellScheduler(); // setup() function for per-TTI algorithm execution void setup(cumac::cumacCellGrpUeStatus* cellGrpUeStatus, cumac::cumacSchdSol* schdSol, cumac::cumacCellGrpPrms* cellGrpPrms, uint8_t in_DL, uint8_t in_columnMajor, uint8_t in_halfPrecision, uint8_t in_lightWeight, cudaStream_t strm); // set in_DL to 1 if setup for DL scheduling; 0 otherwise // in_columnMajor: 0 - row-major channel access, 1 - column-major channel access // in_halfPrecision: 0 - call FP32 floating type kernel, 1 - call FP16 (bfloat162) half-precision kernel // in_lightWeight: 0 - call heavy-weight kernel, 1 - call light-weight kernel // in_enableHarq: 0 - HARQ disabled, 1 - HARQ enabled // requires external synchronization // run() function for per-TTI algorithm execution void run(cudaStream_t strm); // requires external synchronization // parameter/data buffer logging function for debugging purpose void debugLog(); // for debugging only, printing out dynamic descriptor parameters

Multi-cell layer selection

Wrapper class and public member functions:

class cumac::multiCellLayerSel public: // constructor multiCellLayerSel(); // desctructor ~multiCellLayerSel(); // setup() function for per-TTI algorithm execution void setup(cumacCellGrpUeStatus* cellGrpUeStatus, cumacSchdSol* schdSol, cumacCellGrpPrms* cellGrpPrms, uint8_t in_enableHarq, cudaStream_t strm); // in_enableHarq: 0 - HARQ disabled, 1 - HARQ enabled // requires external synchronization // run() function for per-TTI algorithm execution void run(cudaStream_t strm); // requires external synchronization // parameter/data buffer logging function for debugging purpose void debugLog(); // for debugging only, printing out dynamic descriptor parameters

Multi-cell MCS selection + outer-loop link adaptation (OLLA)

Wrapper class and public member functions:

class cumac::mcsSelectionLUT public: // constructor mcsSelectionLUT(uint16_t nActiveUe, cudaStream_t strm); // requires external synchronization // uint16_t

```
nActiveUe is the (maximum) total number of active UEs in all coordinated cells // destructor ~mcsSelectionLUT(); // setup() function for per-TTI algorithm execution void setup(cumacCellGrpUeStatus\* cellGrpUeStatus, cumacSchdSol\* schdSol, cumacCellGrpPrms\* cellGrpPrms, uint8_t in_DL, uint8_t in_baseline, cudaStream_t strm); // in_DL: 0 - UL, 1 - DL // in_baseline: 0 - not using baseline algorithm, 1 - using baseline algorithm // requires external synchronization // run() function for per-TTI algorithm execution void run(cudaStream_t strm); // parameter/data buffer logging function for debugging purpose void debugLog(); // for debugging only, printing out dynamic descriptor parameters
```

Outer-loop link adaptation (OLLA) data structure:

```
// structure containing outer-loop link adaptation algorithm parameters struct ollaParam { float delta; // offset to SINR estimation float delta_ini; // initial value for delta parameter float delta_up; // step size for increasing delta parameter float delta_down; // step size for decreasing delta parameter };
```