第一届5G算法创新大赛

——SCMA

现场宣讲&答疑会

2015年7月16日

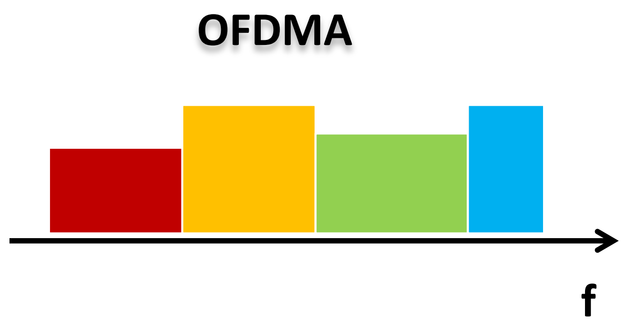
Outline

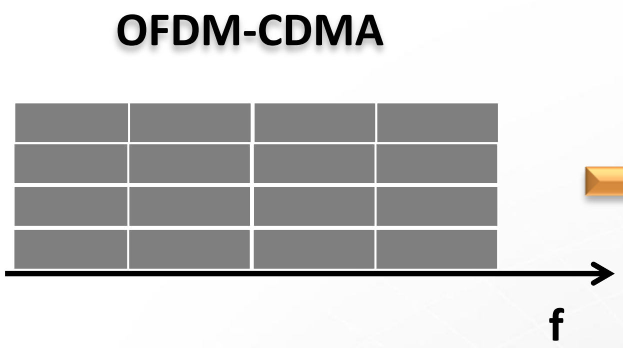
- Warm Up Introductions
- The System You are Going to Implement
- Second Look at SCMA Encoder/Decoder
- Some Other Magic about SCMA
- Real-time SCMA Demo in Field

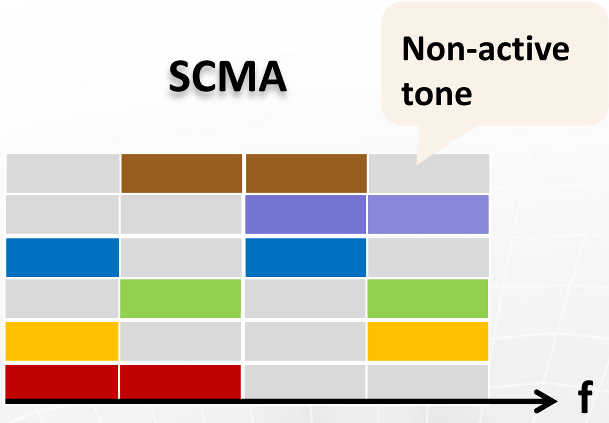
What is SCMA, and why SCMA

WARM UP INTRODUCTIONS

From OFDMA to SCMA







Orthogonal multi-user multiplexing

- Users occupy orthogonal resources for communication
- Easy to implement (single user detection)
- Number of connections limited by the number of physical resource blocks that can be scheduled

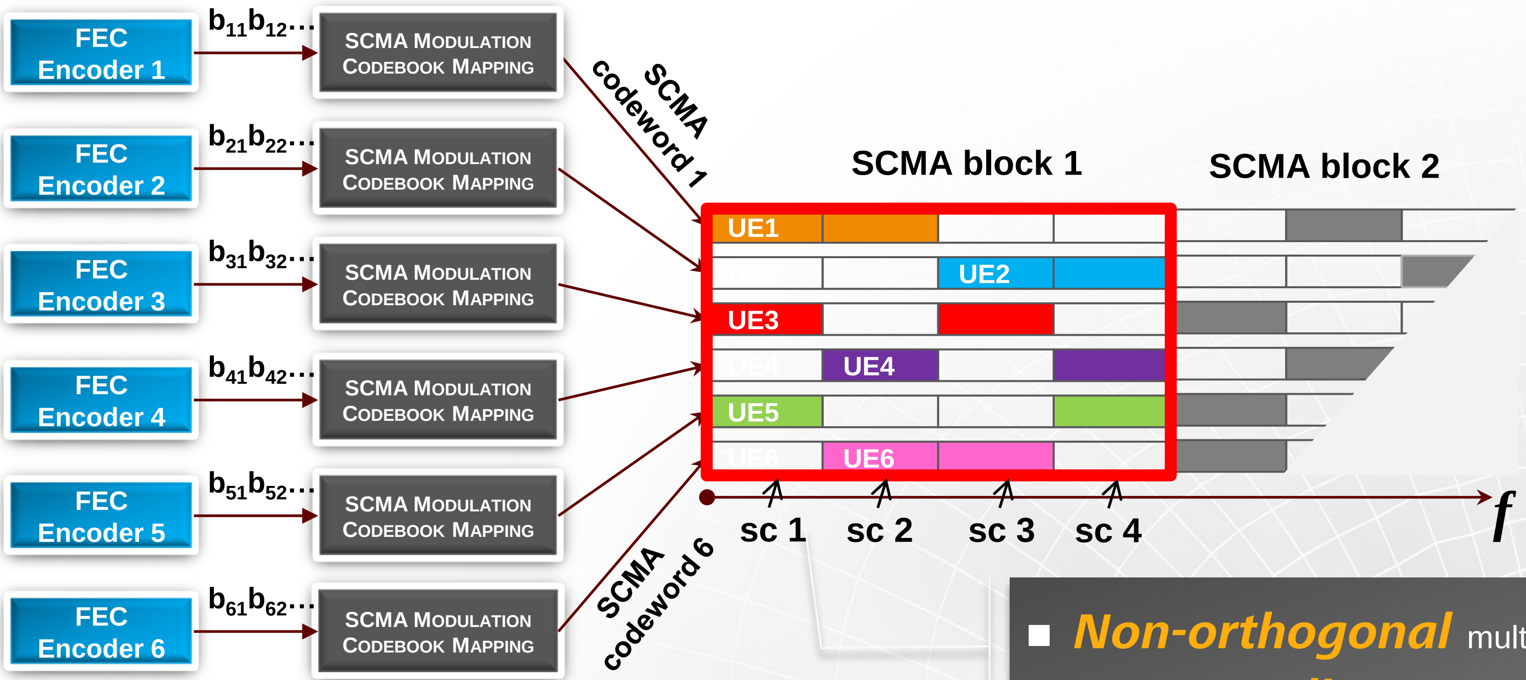
Non-orthogonal multi-user multiplexing

- Users occupy the same resource blocks using CDMA
- Non-practically high multi-user joint detection complexity
- Limited number of concurrent users due to limited sequences
- Better coverage due to spreading gain

Overloaded multi-user multiplexing

- Users occupy the same resource blocks in a low density way
- Affordable low multi-user joint detection complexity
- Less collision even for large number of concurrent Users
- Better coverage due to spreading gain

SCMA (Sparse Code Multiple Access)



A new frequency domain non-orthogonal waveform

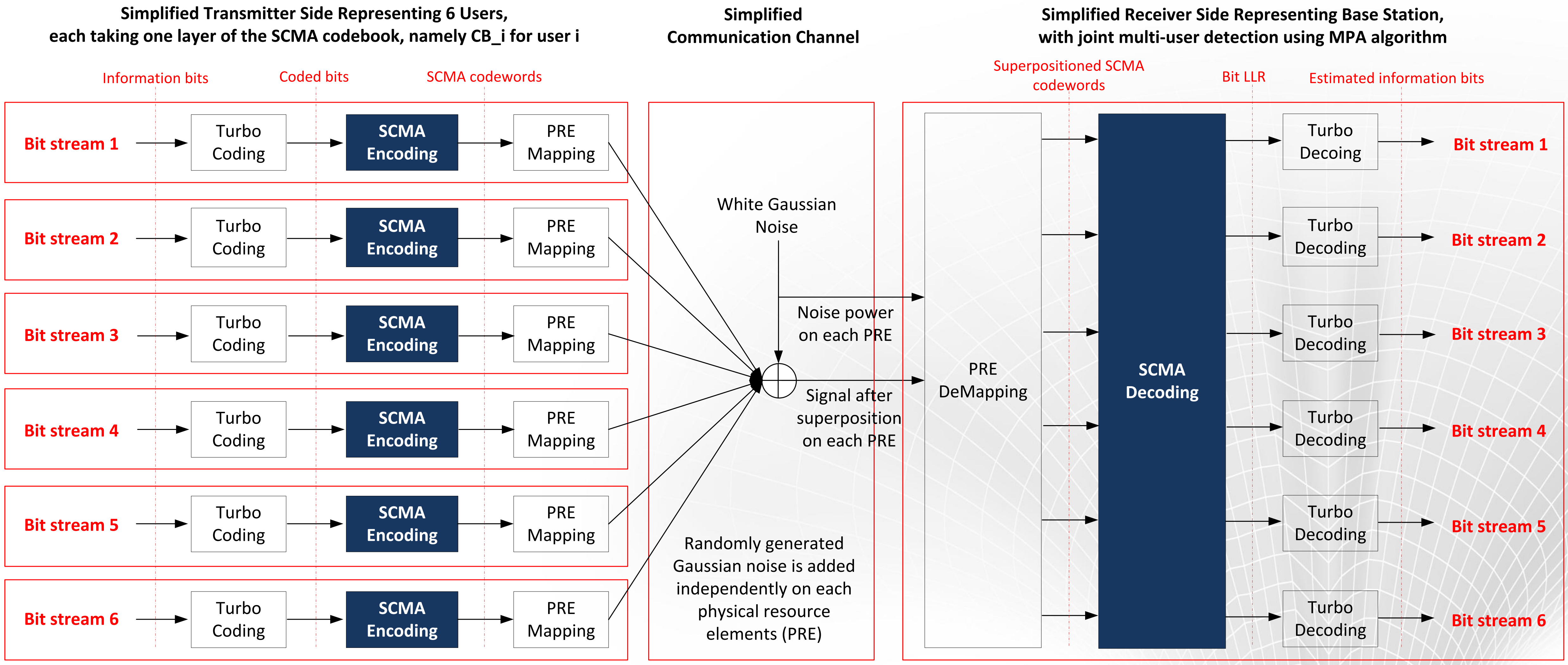
- Input bits are directly mapped to codewords and spread over multiple sub-carriers
- Codewords can be assigned to same UE or different UEs

- ***Non-orthogonal*** multiplexing of code layers
- ***Over-Loading*** to increase overall rate and connectivity
- ***Sparsity*** to limit Rx complexity for detection
- ***Spreading*** for robust link-adaptation, coverage
- ***Multi-dimensional*** codewords with shaping/coding gain

Simplified system, module interface definition, configuration parameters

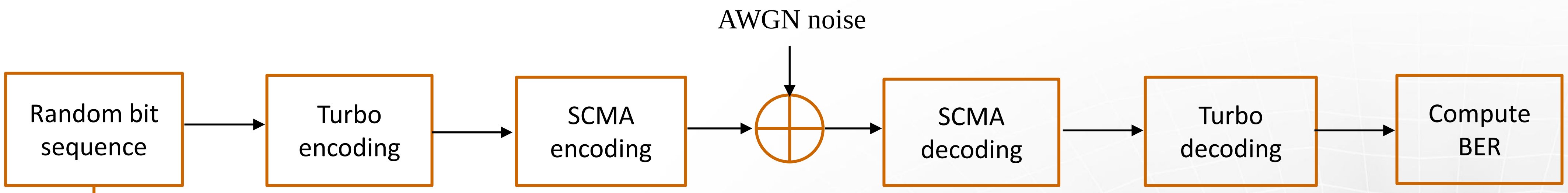
THE SYSTEM YOU ARE GOING TO IMPLEMENT

Simplified Uplink SCMA System to be Implemented



* Turbo coding and decoding can be replaced by other forward error correction (FEC) channel coding/decoding modules.

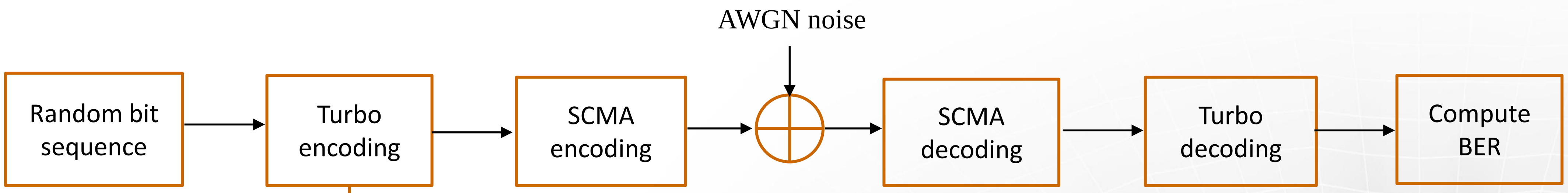
Simplified Uplink SCMA System to be Implemented



- **y = source_bit (B)**
  - **Function： generate uniformly distributed binary sequence.**
  - **Input：**
    - **B – length of the binary sequences.**
  - **Output： binary sequences with length B.**

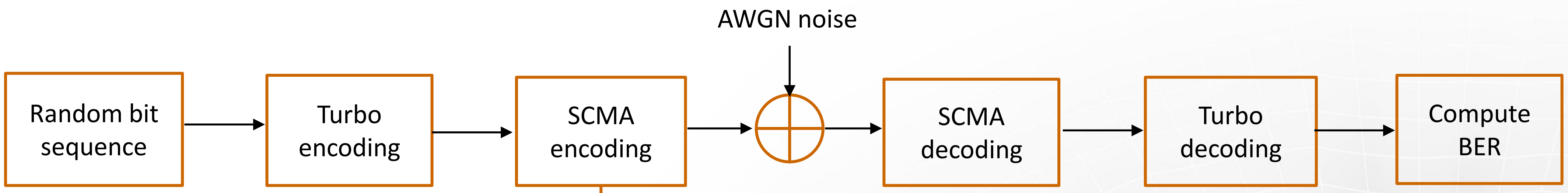
Note: Every module is performed for data streams of 6 users.

Simplified Uplink SCMA System to be Implemented



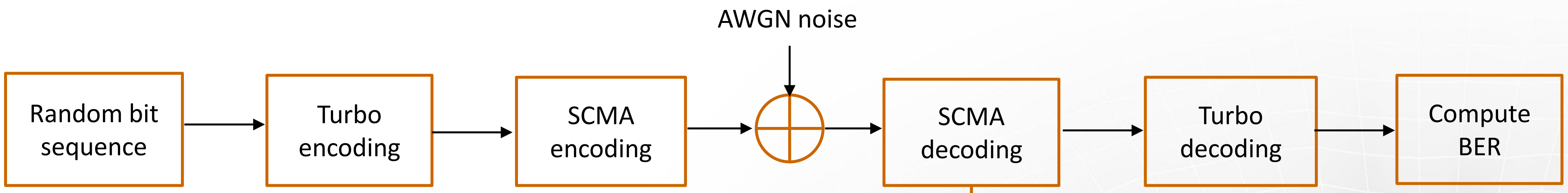
- **y = turbo_encoding(source_bit, R)**
- **Function： to perform Turbo encoding.**
- **Input：**
  - **source_bit – binary sequence to be encoded;**
  - **R – coding rate.**
- **Output： Turbo encoded bits with length N = B/R.**

Simplified Uplink SCMA System to be Implemented



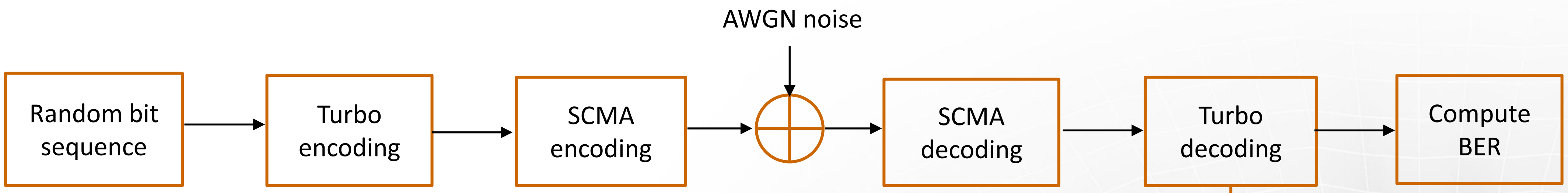
- **y = scma_encoding(encode_seq, CB)**
  - **Function： bit sequences from 6 users are SCMA encoded according to specified codebooks.**
  - **Input：**
    - **encode_seq – encoded bit streams of 6 layers;**
    - **CB – codebooks for each layer.**
  - **Output： superposed symbols to be transmitted.**

Simplified Uplink SCMA System to be Implemented



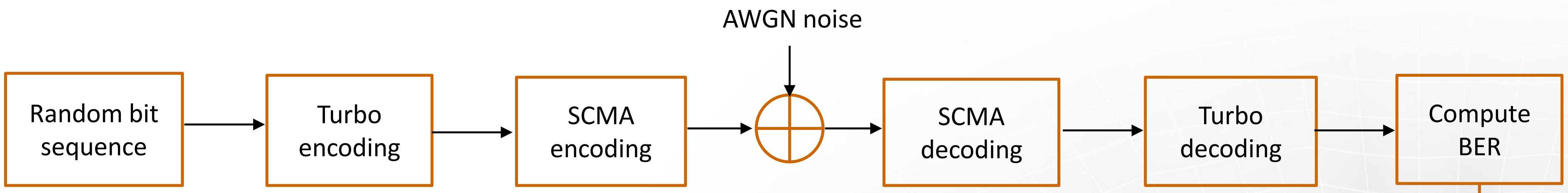
- **y = scma_decoding(r, CB, H, N0)**
- **Function： to recover the bit information of each layer**
- **Input：**
  - **r – received signals from the channel;**
  - **CB – codebooks of each layer;**
  - **H – channel state information;**
  - **N0 – noise information.**
- **Output： LLRs of encoded bits.**

Simplified Uplink SCMA System to be Implemented



- **y = turbo_decoding(LLR, rate)**
  - **Function： to perform Turbo decoding.**
  - **Input：**
    - **LLR – bit LLRs output from SCMA decoder;**
    - **rate – channel coding rate.**
  - **Output： recovered source bit sequence.**

Simplified Uplink SCMA System to be Implemented



- y = err_check(source_bit, decode_bit)
  - **Function： to calculate the BLER**
  - **Input：**
    - **source_bit – source bits of each user;**
    - **decode_bit – recovered bits of each user;**
  - **Output： average BER.**

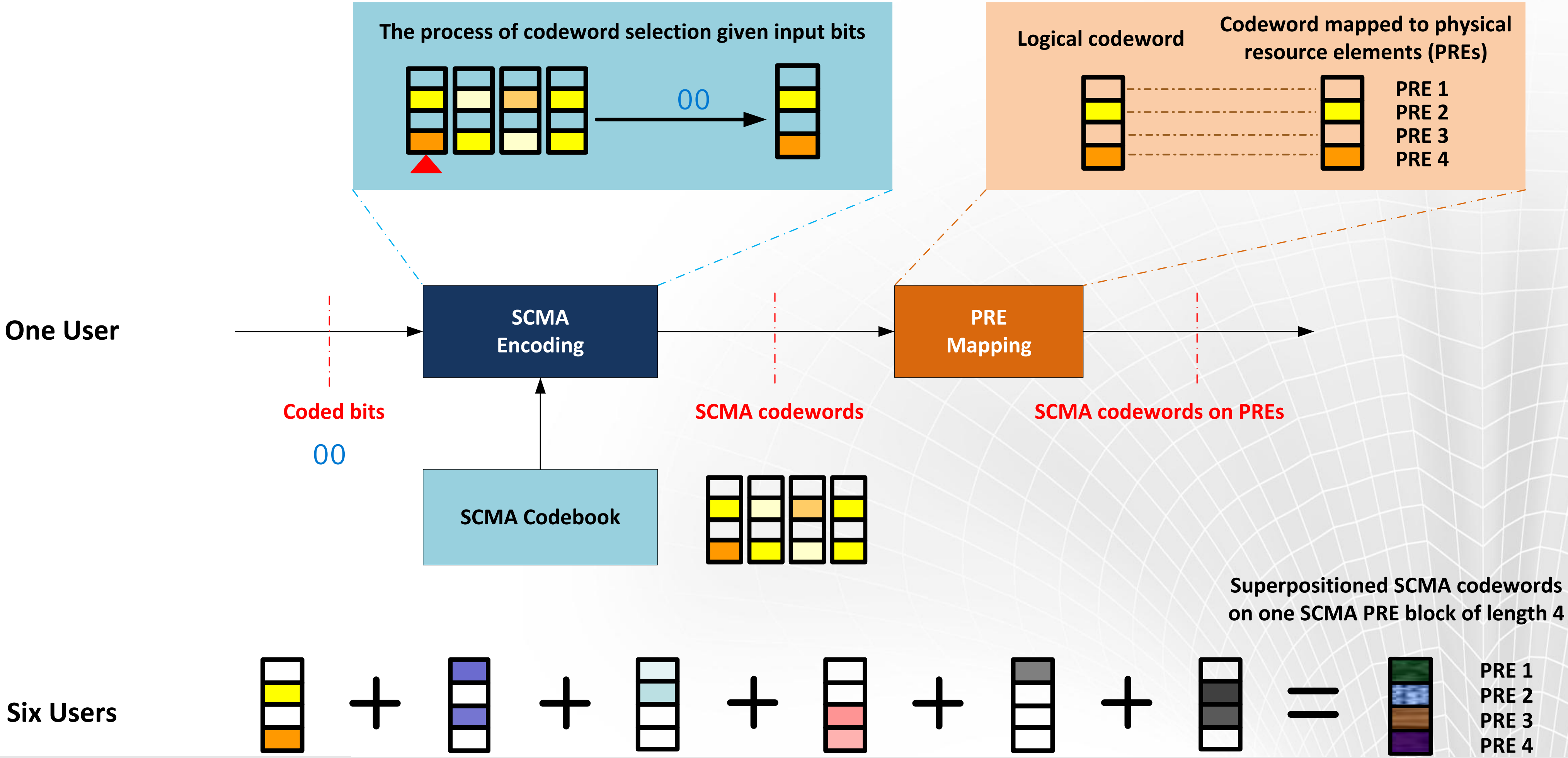
System Configuration Parameters for Implementation

Parameter Categories	Related Variables	Typical value	Description
SCMA Codebook	V	6	6 variable nodes (VN), number of data layers
	F	4	4 function nodes (FN), number of physical resources
	d_f	3	Each FN is connected to 3 VNs
	d_v	2	Each VN is connected to 2 FNs
	M	4	Number of codeword in each codebook
	CB_i	F-by-M matrix	Codebook for one SCMA data layer, given
Turbo Coding	R	1/2	Coding rate, defined as the ratio of information bits over coded bits
SCMA decoding	N_iter	3 ~ 15	Number of iterations in MPA
	H_n	{1}	Channel gain, in the white Gaussian noise only case, H_n={h_n,k}={1}
	APP_i	1/M	A prior probability of codeword i, assuming equal probability 1/M
System Scale	B	125 bytes = 1000 bits	Total number of information bits, randomly generated
	N	B / R = 2000 bits	Total number of coded bits after Turbo coding
	L	L = F * N/log2(M) = 4000	Total number of physical resource units

MPA Algorithm, complexity reduction hints

SECOND LOOK AT SCMA ENCODER/DECODER

How to Do SCMA Encoding with SCMA Codebook

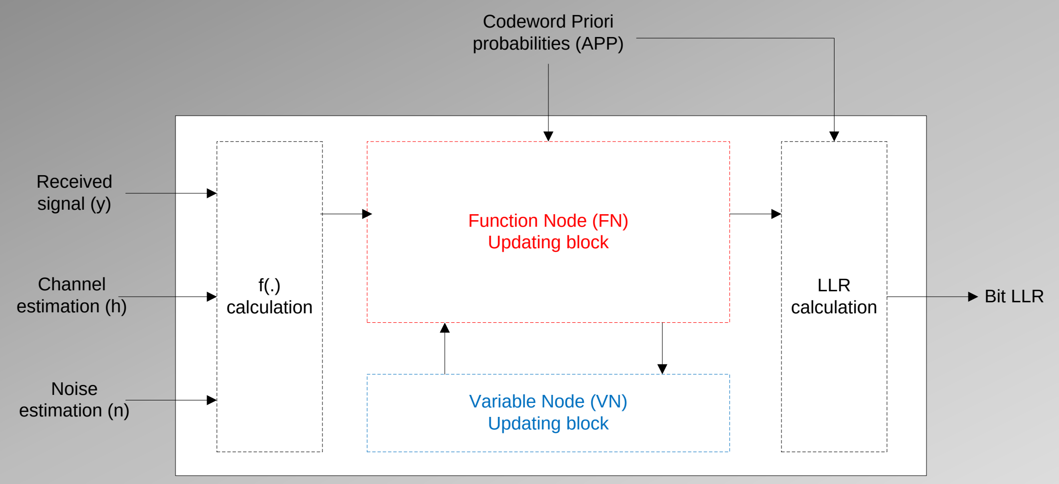


How to Do SCMA Decoding with MPA Algorithm

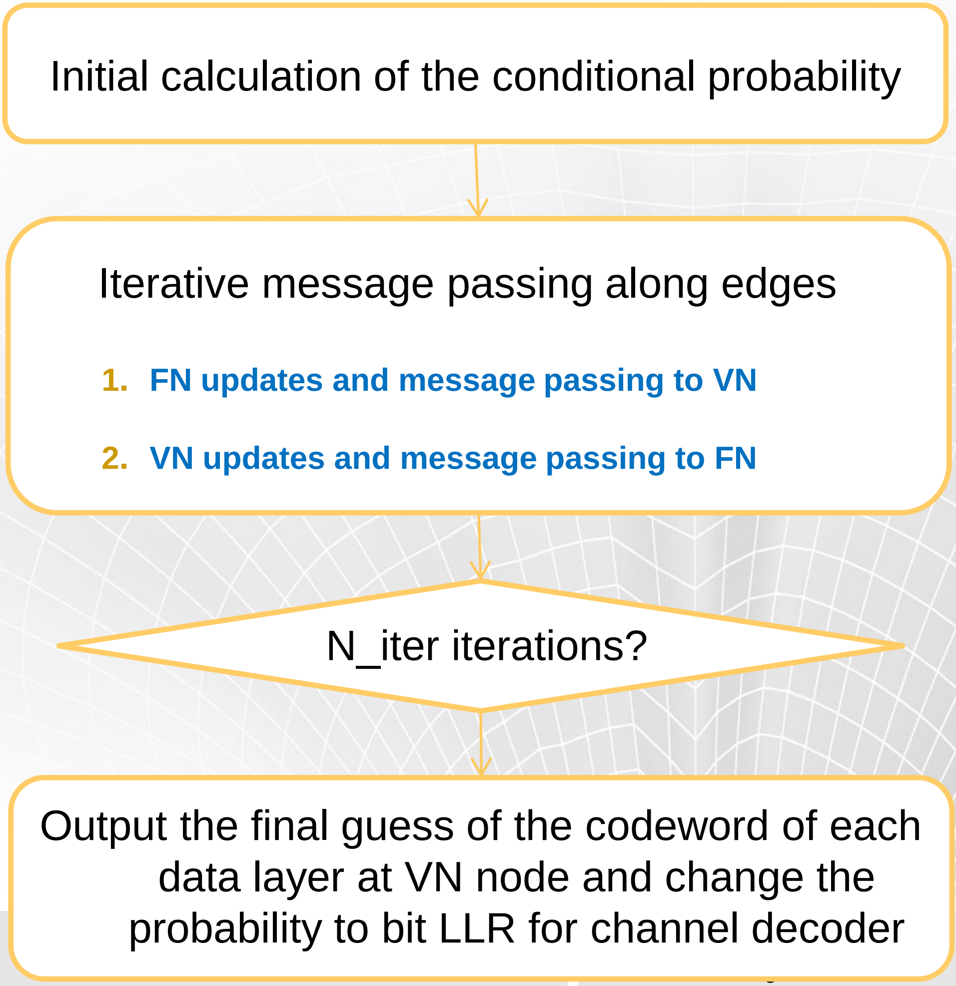
Selection of SCMA Decoder

- The optimal multi-user detection can be done by using the maximum joint a posteriori probability (MAP) detection with excessive search – non-practical complexity
- With the low density spreading structure employed in SCMA, we can derive near ML performance multi-user detection with message passing algorithm (MPA) – affordable complexity

Diagram for Massage Passing Algorithm

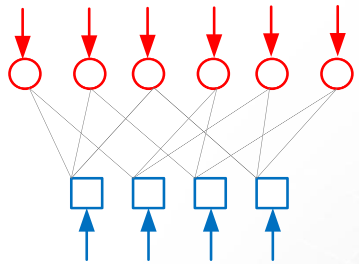


MPA Decoder (Performed for each SCMA block)



How to Do SCMA Decoding with MPA Algorithm

Parameters	Description of the parameters
y_n, n=1,…, F	Received signal as input to the MPA decoder on resource n
m_k, k=1,…, V	Codeword selected by layer k, m_k = 1,…, M
No_n, n=1,…, F	Noise power estimation on physical resource n
C_k,n(m_k)	The constellation symbol of VN node k on physical resource n when using codeword m_k
H_n = {h_n,k}	Channel gain of user k on physical resource n
Ap_k, k=1,…, V	A prior probability of codeword k, assuming equal probability 1/M
LLR_k,b	logarithm of the likelihood ratio of layer k bit b
N_iter	Number of iterations in the MPA



Step 1: Initial calculation of the conditional probability

- For each function node FN, calculate the f_n() function, which is the set of all possible residual signals given the known or estimated channel h_n,k and the assumed transmitted codeword C_k,n(m_k)
- When d_f = 3, as in the example, for each FN node n, there are M*M*M combinations of transmitted signals, so there are in total F*M*M*M values to store for f() function calculation

$$f_n(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n) = \frac{-1}{N_{0,n}} \left\| y_n - \left(h_{n,1}\mathbf{C}_{1,n}(m_1) + h_{n,2}\mathbf{C}_{2,n}(m_2) + h_{n,3}\mathbf{C}_{3,n}(m_3) \right) \right\|^2$$

$$m_1 = 1, \dots, M \quad m_2 = 1, \dots, M \quad m_3 = 1, \dots, M \quad n = 1, \dots, F$$

- Phi_n() function is actually the conditional probability for given codeword combination, for Gaussian noise case, it is the exponential operation over f_n function, so the storage needed is the same

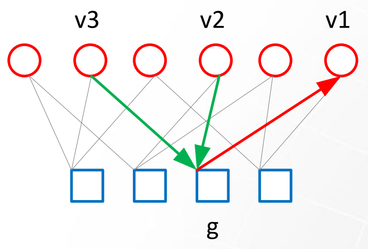
P(yn|x1, x2, x3) ----- $\Phi_n(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n) = \exp\left(f_n(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n)\right)$

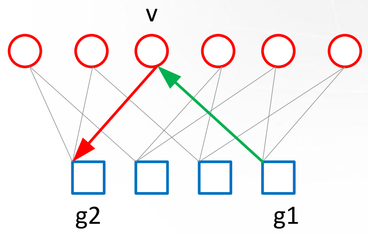
- To prepare for the iterations, we assign the a prior probability for each codeword, which is assumed to be equal

P(x1), P(x2), P(x3)---- $I_{v_1 \to g}^{init}(m_1) = I_{v_2 \to g}^{init}(m_2) = I_{v_3 \to g}^{init}(m_3) = \frac{1}{M}$

How to Do SCMA Decoding with MPA Algorithm

Parameters	Description of the parameters
y_n, n=1,…, F	Received signal as input to the MPA decoder on resource n
m_k, k=1,…, V	Codeword selected by layer k, m_k = 1,…, M
No_n, n=1,…, F	Noise power estimation on physical resource n
C_k,n(m_k)	The constellation symbol of VN node k on physical resource n when using codeword m_k
H_n = {h_n,k}	Channel gain of user k on physical resource n
Ap_k, k=1,…, V	A prior probability of codeword k, assuming equal probability 1/M
LLR_k,b	logarithm of the likelihood ratio of layer k bit b
N_iter	Number of iterations in the MPA





Step 2: Iterative message passing along edges

[FN update]: message passing from FN to its neighboring VNs

- FN node g passes updates obtained from extrinsic information to its neighboring VN nodes (g to v1, information from v2 and v3 are extrinsic)
- The message passed to v1 contains the guess of what signal at g may be given all possibilities of v1

$$I_{g\to v_1}(m_1) = \sum_{m_2=1}^{M}\sum_{m_3=1}^{M} \phi_n\left(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n\right)\left(I_{v_2\to g}(m_2) I_{v_3\to g}(m_3)\right) \quad m_1 = 1, \dots, M$$

$$I_{g\to v_2}(m_2) = \sum_{m_1=1}^{M}\sum_{m_3=1}^{M} \phi_n\left(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n\right)\left(I_{v_1\to g}(m_1) I_{v_3\to g}(m_3)\right) \quad m_2 = 1, \dots, M$$

$$I_{g\to v_3}(m_3) = \sum_{m_1=1}^{M}\sum_{m_2=1}^{M} \phi_n\left(y_n, m_1, m_2, m_3, N_{0,n}, \mathbf{H}_n\right)\left(I_{v_1\to g}(m_1) I_{v_2\to g}(m_2)\right) \quad m_3 = 1, \dots, M$$

[VN update]: message passing from VN to its neighboring FNs

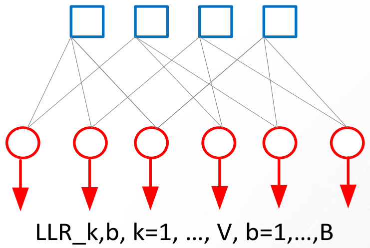
- VN node v passes updates obtained from extrinsic information to its neighboring FN nodes (v to g2, information from g1 is extrinsic)
- In the dv=2 case, it is actually a “guess” swap at VN node

$$I_{v\to g_1}(m) = \text{normalize}\left(ap_v(m)\, I_{g_2\to v}(m)\right) \quad m = 1, \dots, M$$

$$I_{v\to g_2}(m) = \text{normalize}\left(ap_v(m) I_{g_1\to v}(m)\right) \quad m = 1, \dots, M$$

How to Do SCMA Decoding with MPA Algorithm

Parameters	Description of the parameters
y_n, n=1,…, F	Received signal as input to the MPA decoder on resource n
m_k, k=1,…, V	Codeword selected by layer k, m_k = 1,…, M
No_n, n=1,…, F	Noise power estimation on physical resource n
C_k,n(m_k)	The constellation symbol of VN node k on physical resource n when using codeword m_k
H_n = {h_n,k}	Channel gain of user k on physical resource n
Ap_k, k=1,…, V	A prior probability of codeword k, assuming equal probability 1/M
LLR_k,b	logarithm of the likelihood ratio of layer k bit b
N_iter	Number of iterations in the MPA



Step 3: LLR output at variable node after N_iter iterations

- After N_iter iterations, we shall output the guess at each VN node (for each data layer) as the detection results
- The guess at VN node v for codeword m is a chain product of all guesses from all its neighboring FN nodes and the a prior probability

$$Q_v(m) = ap_v(m)\, \mathbf{I}_{g_1 \to v}(m)\, \mathbf{I}_{g_2 \to v}(m) \qquad m = 1, \ldots, M$$

- After getting the probability guess of codeword at each layer, we then need to calculate the Log-Likelihood-Rate (LLR) for each coded bit, so that they can serve as the input for the turbo decoder (or any other channel decoder) directly after MPA

$$LLR_x = \log\left(\frac{P(b_x = 0)}{P(b_x = 1)}\right)$$

$$LLR_x = \log\left(\frac{\sum_{\mathrm{m:b_{m,x}=0}} Q_v(\mathrm{m})}{\sum_{\mathrm{m:b_{m,x}=1}} Q_v(\mathrm{m})}\right) = \log\left(\sum_{\mathrm{m:b_{m,x}=0}} Q_v(\mathrm{m})\right) - \log\left(\sum_{\mathrm{m:b_{m,x}=1}} Q_v(\mathrm{m})\right)$$

Hints on Low Complexity MPA Receiver Design

- **Short-comings of the current MPA algorithm**
  1. Though much lower complexity compared with the optimal MAP algorithm (thanks to the sparse structure of the SCMA codebook), it is still of high complexity for hardware
  2. The exp(.) operations causes very large dynamic ranges and very high storage burden if using lookup table, which is not good news for hardware implementation

- **Hint 1: Change to LOG domain using Jacobi's logarithm**
  1. After changing to Log domain, exp(.) operation disappears : MPA -> MAX-Log MPA

$$\log\left(\sum_{i=1}^{N}\exp(f_i)\right) \approx \max\{f_1, f_2, \ldots, f_N\}$$

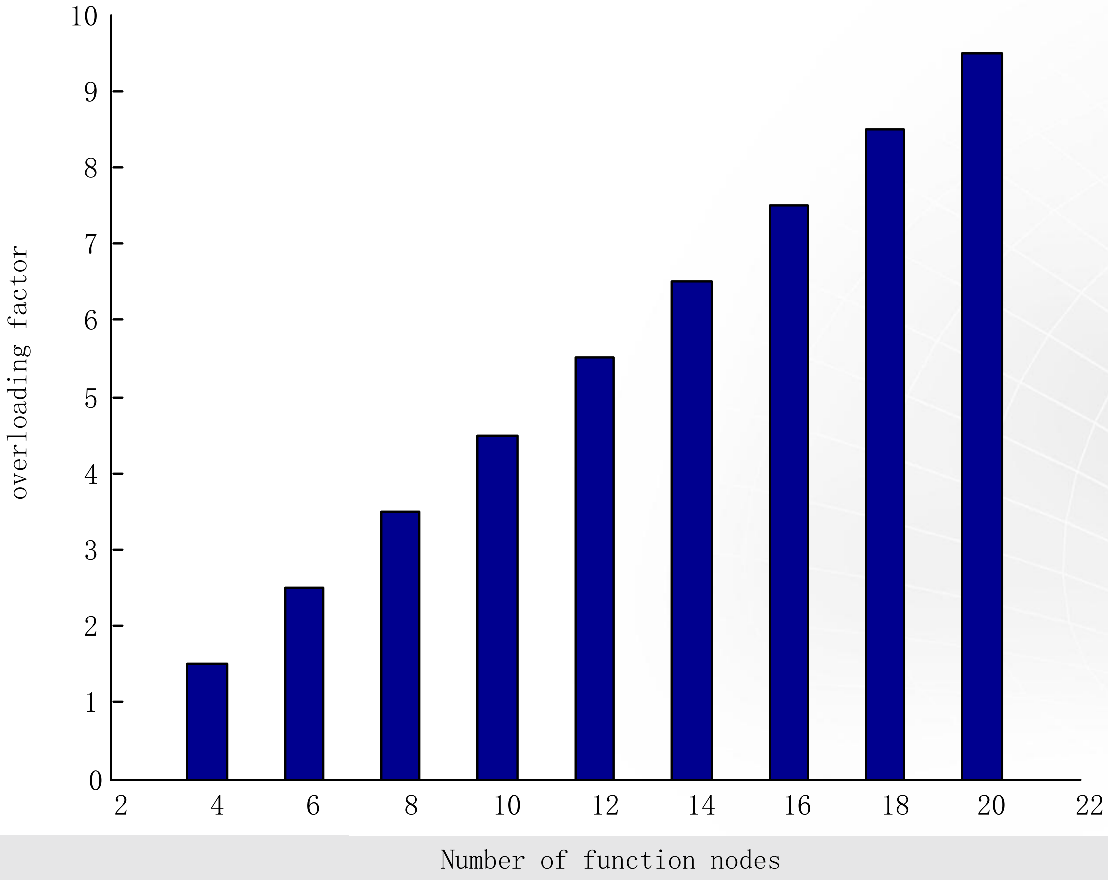
- **Hint 2: Optimize the calculations during iterations**
  1. Try to optimize the order of iterations
  2. Try to use as much as possible the common results in the calculation

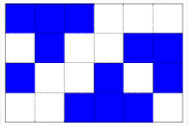
flexible overloading, Grant-free, Outer-loop Turbo MPA

SOME OTHER MAGIC OF SCMA

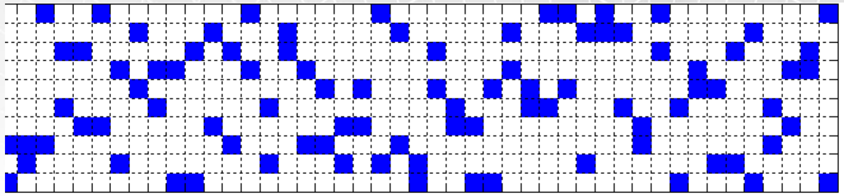
Flexible overloading to support massive connectivity

$$\text{overloading factor} = \frac{\text{UEs/data layers}}{\text{number of resources}}$$



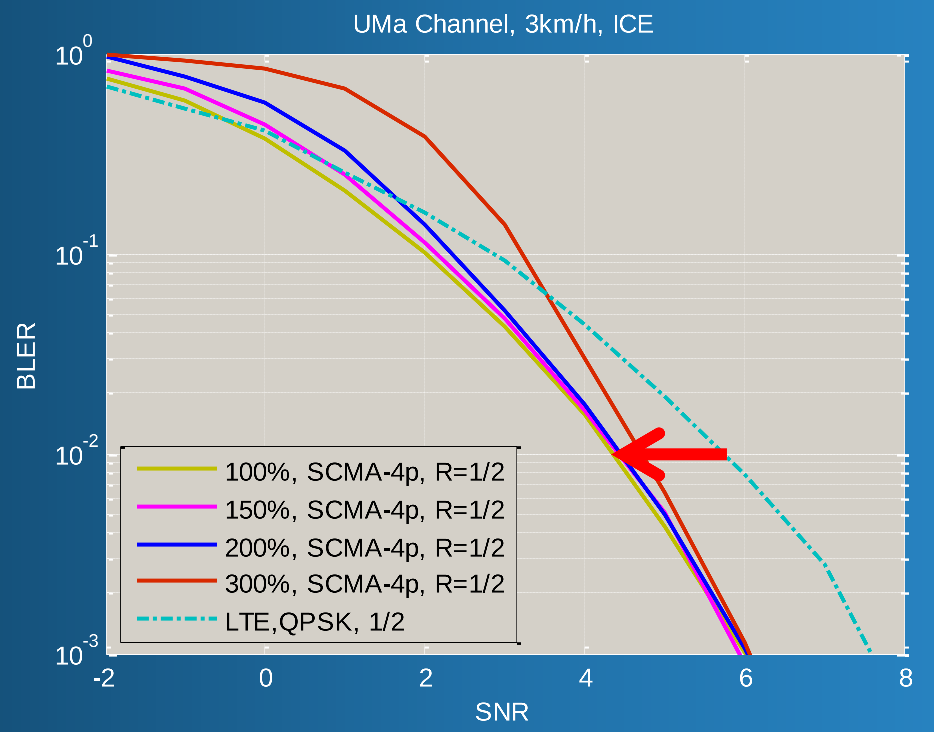


FN:10; VN:45; OV=450%



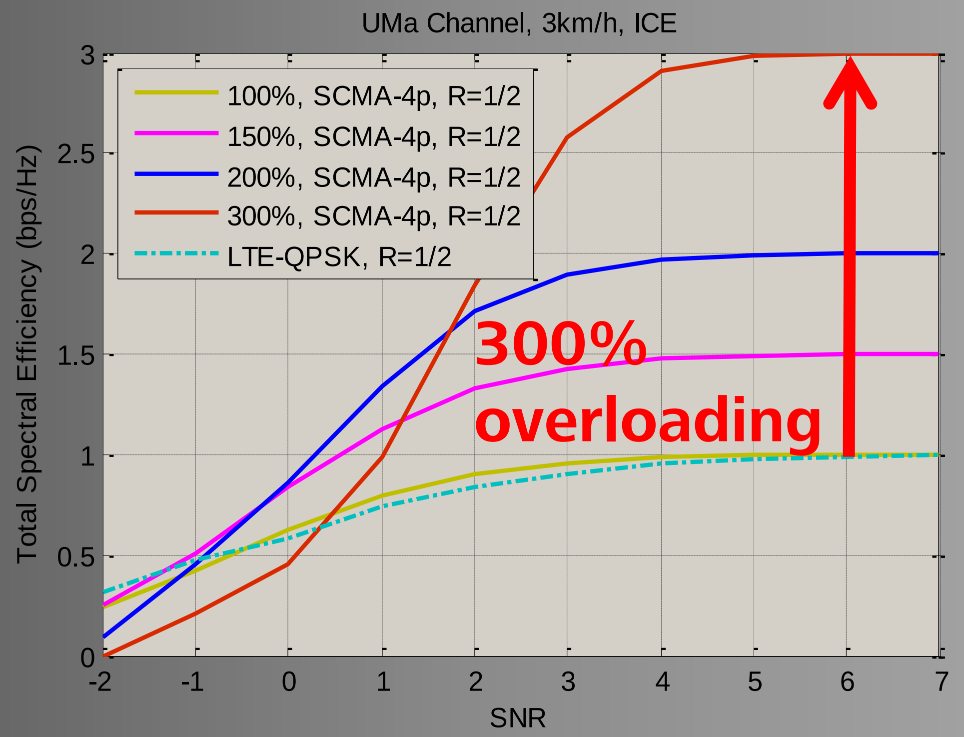
Flexible overloading to support massive connectivity

Better Link Quality over LTE



- SCMA has SNR gain over LTE (same rate & power per user)
- SCMA with overloading performs towards single user

300% Larger Number of Connected Users



- **Given the same SNR, SCMA can boost total system throughput up to 300% over LTE (@LTE BLER=0.01)**

SCMA Offers Better Link Quality and 300% Larger Number of Physical Link Connections over LTE.

Blind detection to support Grant-free transmission

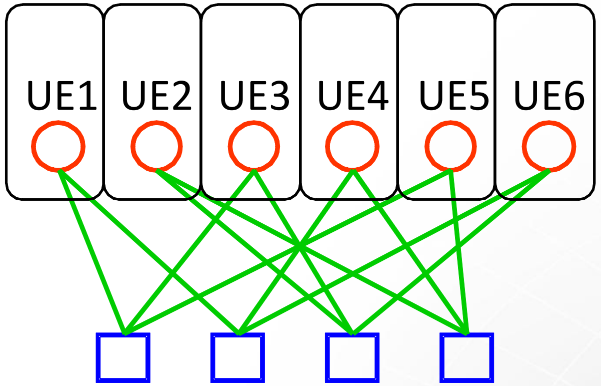
Blind detection

- Joint detection of users' status and data;
- Robust to codebook collision.

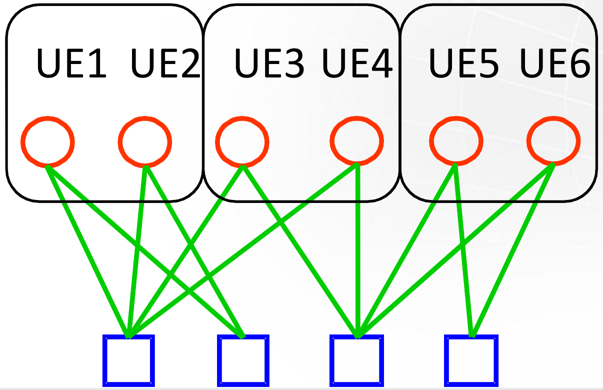
Grant-free transmission

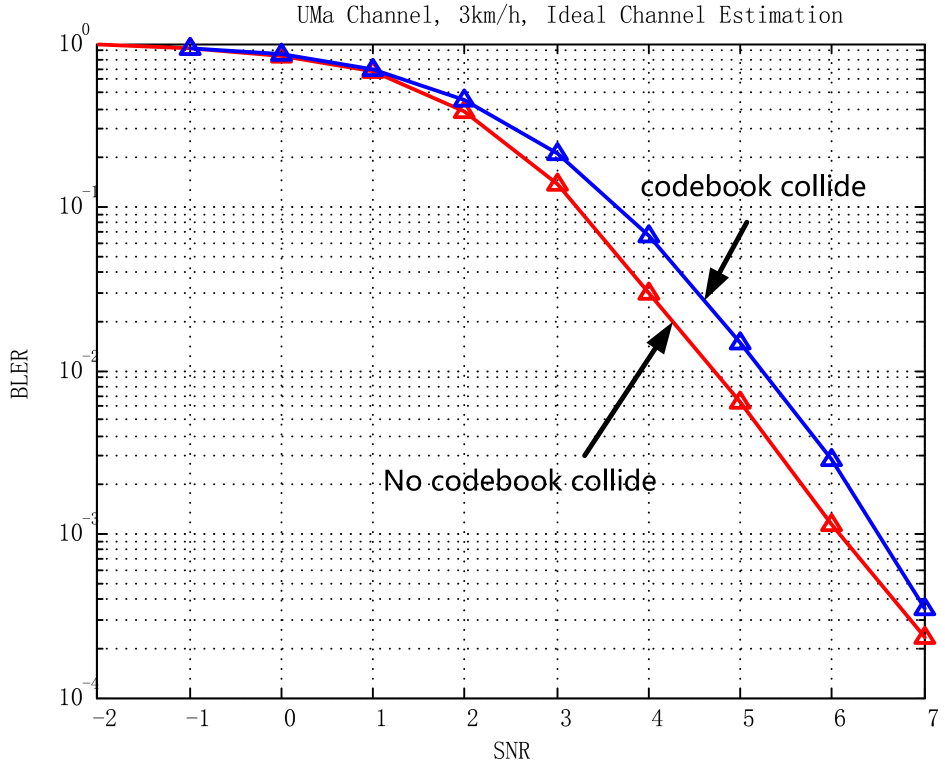
- No need to schedule users for uplink transmissions.

no codebook collide



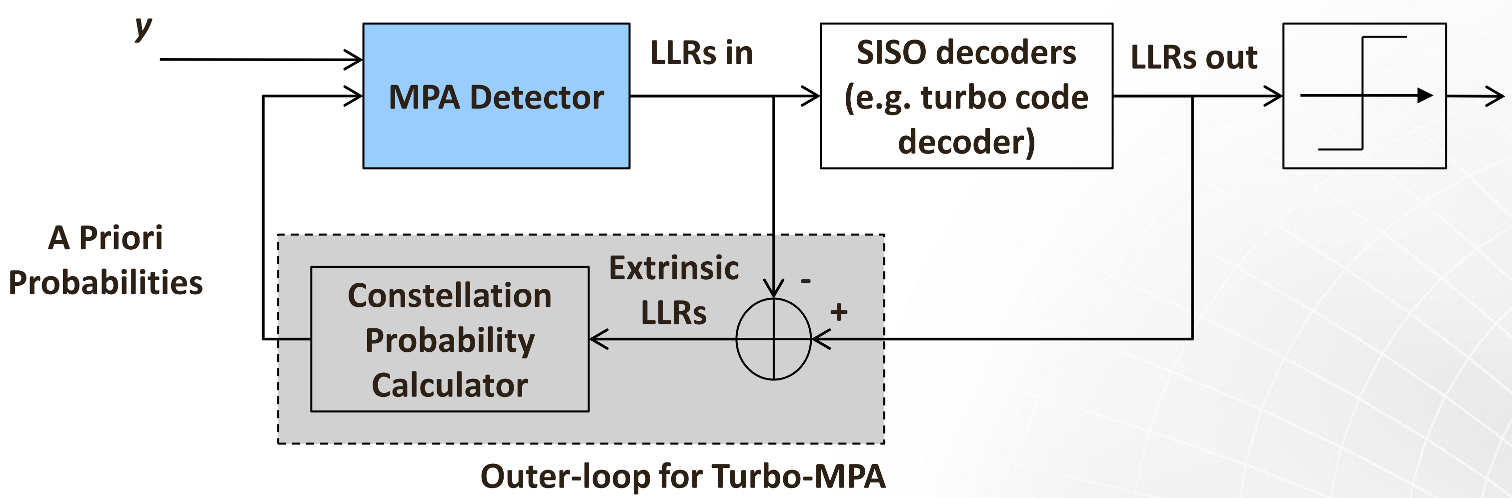
codebook collide

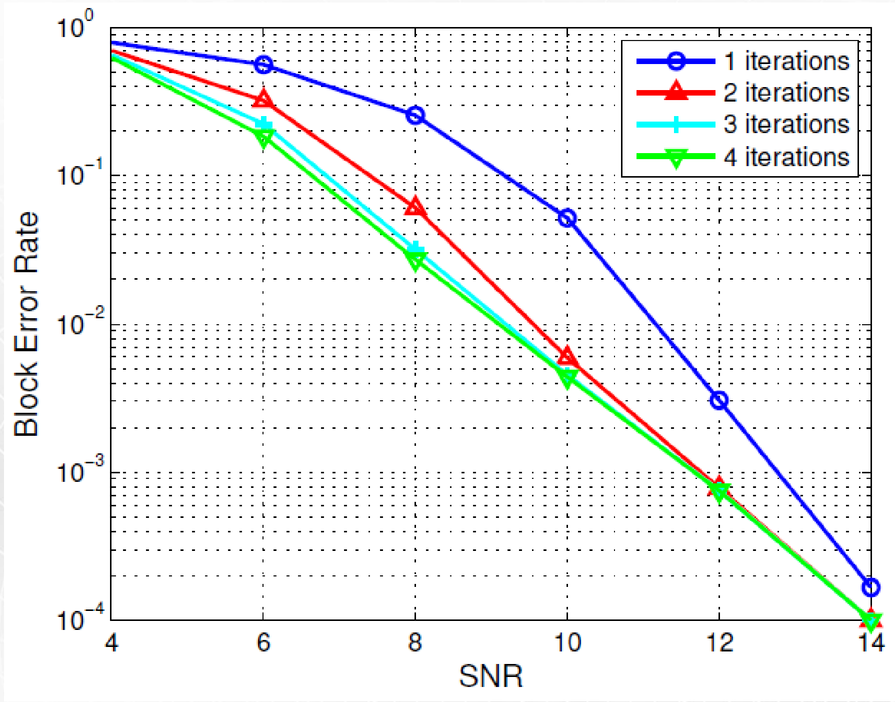




when the codebook collides, the performance loss is less than 0.5dB

Outer-loop Turbo MPA to enhance the performance

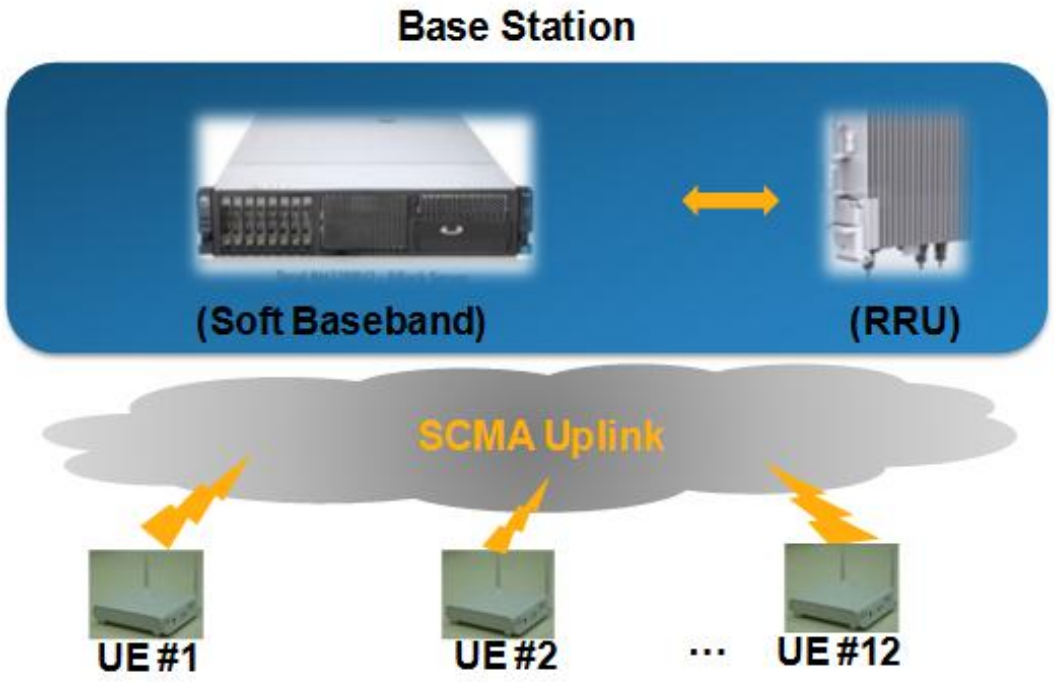




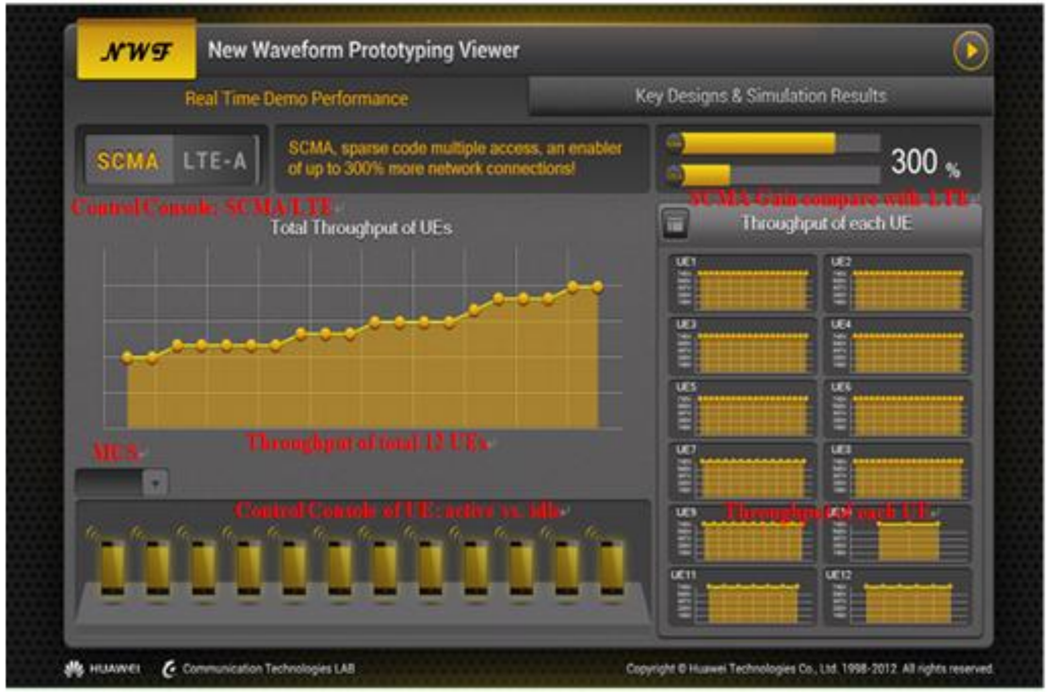
The iterative detection between MPA decoder and Turbo decoder can significantly improve the performance.

about 2 dB SNR gain at the BLER of 10^{-2} with 4 iterations

REAL-TIME SCMA DEMO IN FIELD



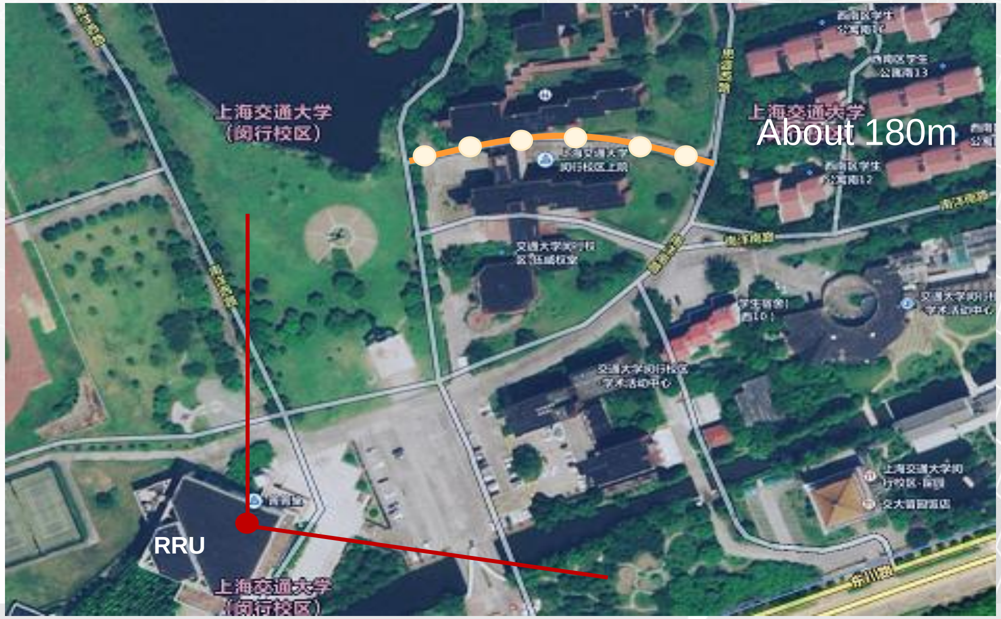
Parameters	Contents
Frame Structure	TDD, Uplink, LTE R10
Mode	SCMA over F-OFDM
Carrier Frequency	2.6GHz
System Bandwidth	20MHz
No. of UEs @4RB (20 byte small packets for each UE)	LTE baseline: 4 SCMA: up to 12
Total UE Tx Power	23dBm (Max.)
Rx Antenna Conf. of eNB	2 Rx
Tx Antenna Conf. of UE	1 Tx



Case 1.1: 6 ue in same fixed location

Case 1.2: 6 ue in different fixed location

Case 1.3: 6 ue walk around road (near)



Case 1.4: 6 ue walk around road (far)

THANK YOU