User Clustering as Optimization Problem in Real-Time Resource Management

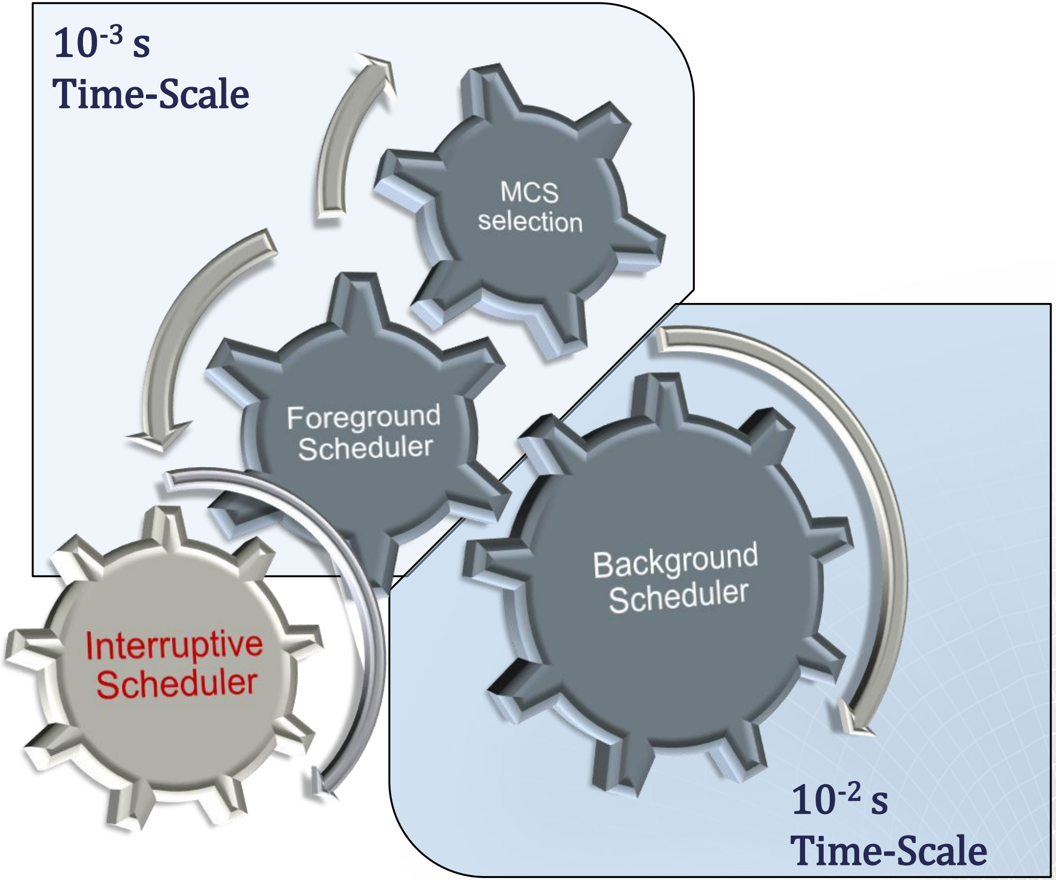
Vladimir Lyashev

RTT Algorithm Technology Lab.

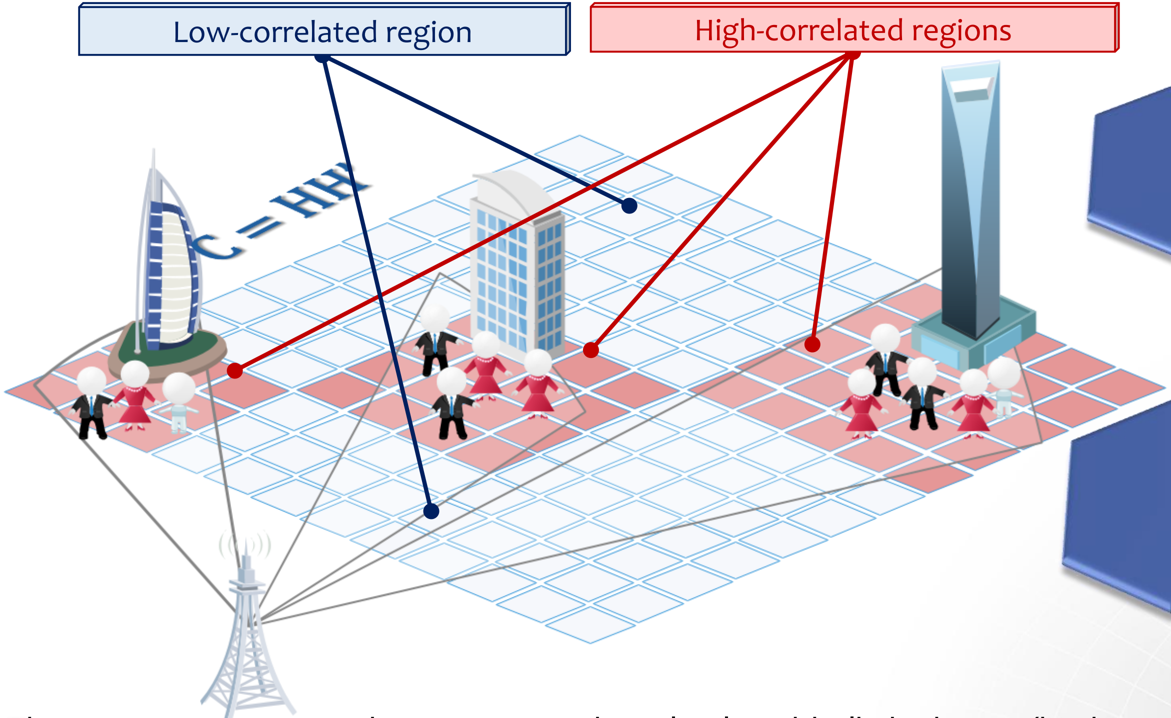
Moscow Research Center

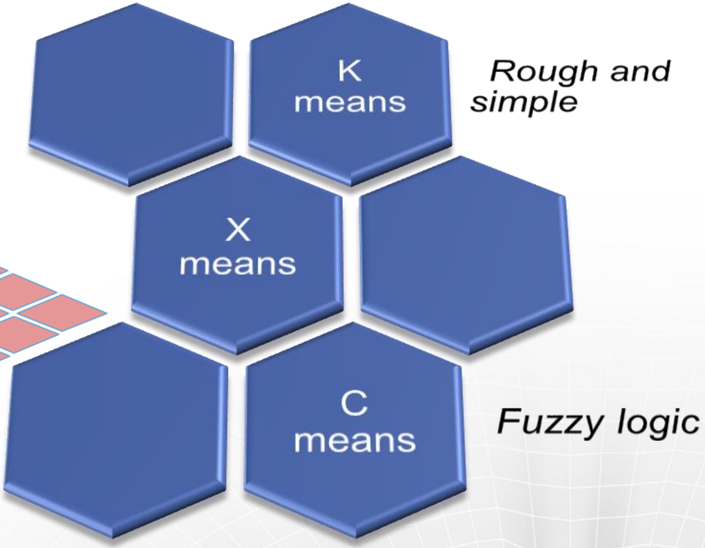
HUAWEI | MOSCOW RESEARCH CENTER

Scheduler Slicing Architecture



Users clustering





There are many cases when users are densely placed in limited area (business centers, hotels, bus stations, etc) which makes possible to GROUP them by spatial or channel criteria and provide simple pairing – pick up a single user from each high-correlated group and pair him with users from another groups!

USERS CLUSTERING PROBLEM

Clustering as optimization problem

Given a set of observations $(x_1, x_2, , x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k \leq n$ sets $S = S_1, S_2, , S_k$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_S \sum_{i=1}^{k} |S_i| \operatorname{var} S_i \to \arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2,$$

where μ_i is the mean of points in S_i.

From geometry to channel correlation

Because the total variance is constant, this is also equivalent to maximizing the squared deviations between points in different clusters (between-cluster sum of squares).

However, it is not unique criteria for cluster description. In our task formulation, we can define another criteria that is more relevant to correlation properties of the user channels.

Let describe space properties of the channel by its main eigenvector $\mathbf{v}_q$ for user q. Then it is possible to introduce correlation coefficient $c_{pq} = \mathbf{v}_q^H \mathbf{v}_p$, which shows spatial correlation between users p and q. In this case we can make new definition for k-mean techniques:

$$\arg\min_{S} \sum_{i=1}^{k} |S_i| \text{ var } S_i \to \arg\max_{S} \sum_{i=1}^{k} \sum_{q \in S_i} \dot{c}_q,$$

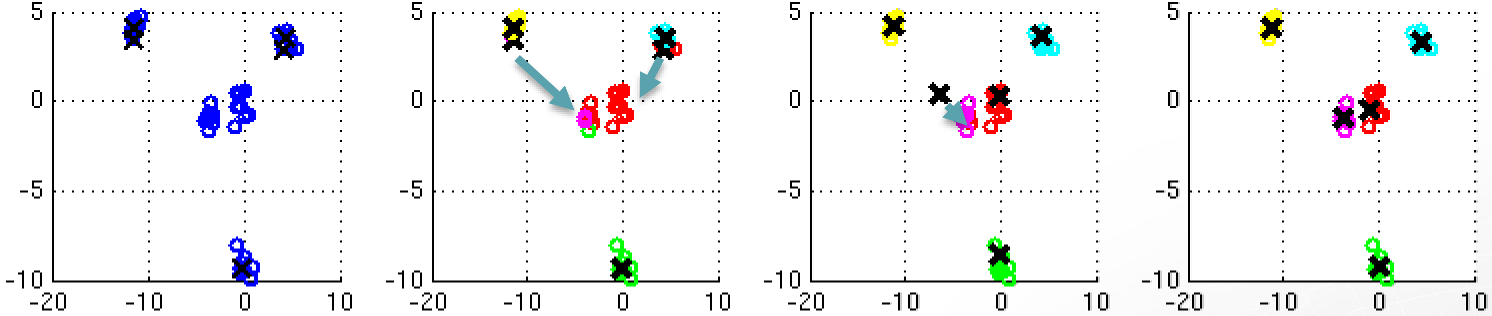
where $\dot{c}_q = \mathbf{v}_{S_i}^H \mathbf{v}_q$ - correlation between user eigenvector and centroid eigenvector, and centroid eigenvector can be defined i.e.

$$\mathbf{v}_{S_i} = \frac{\sum_{q \in S_i} \mathbf{v}_q}{\left\| \sum_{q \in S_i} \mathbf{v}_q \right\|}.$$

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as *Lloyd's algorithm*, particularly in the computer science community.

K-means

- Fixed number of centroids
- Every iteration centroid is updating as mean of corresponding cluster points
- Stop when clusters don’t change anymore



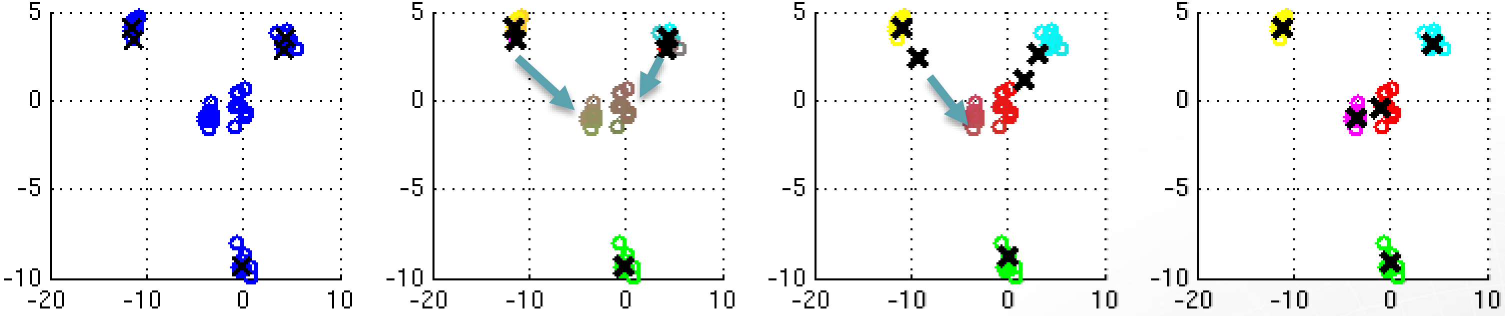
k-means	+ Acceptable complexity
	+ Guaranteed clustering (uniform)
	- Low flexibility

$$\arg\min_{S} \sum_{j=1}^{C} \sum_{x_i \in S_j} \left\| x_i - \mu_j \right\|^2 ,$$

$$\mu_j = \frac{1}{|S_j|} \sum_{x \in S_j} x_i .$$

C-means

- Fixed number of centroids
- Every dot owns probability of being assigned to every centroid
- Every iteration centroid is updating as mean of corresponding cluster points weighted by probability of assigning
- Stop when clusters don't change anymore or changes of probabilities are less than a threshold



c-means

- \+ Acceptable complexity
- \+ High after-clustering versatility
- \- Low flexibility of cluster number selection

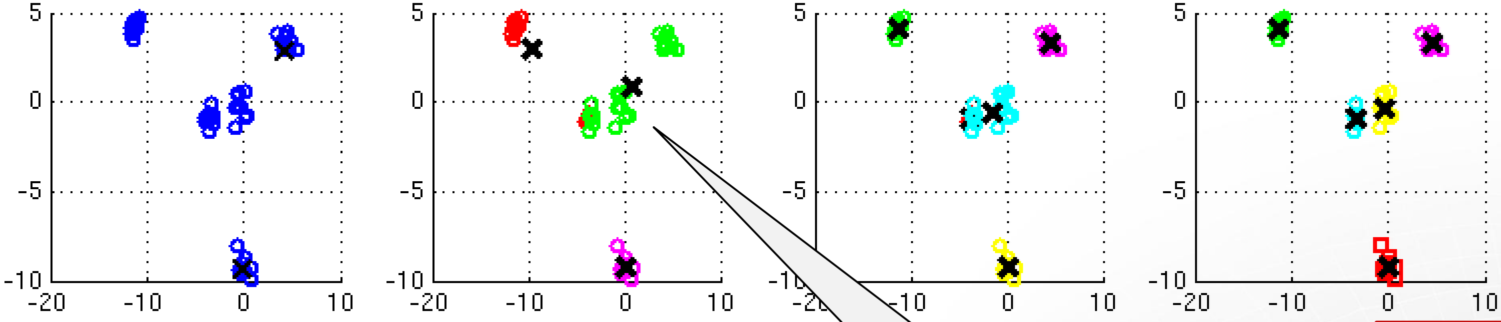
$$\arg\min_{\mu} \sum_{j=1}^{C} \sum_{i=1}^{N} u_{ij}^{m} \left\| x_i - \mu_j \right\|^2 ,$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - \mu_j \right\|^2}{\left\| x_i - \mu_k \right\|^2} \right)^{\frac{2}{m-1}}} , \quad \mu_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} .$$

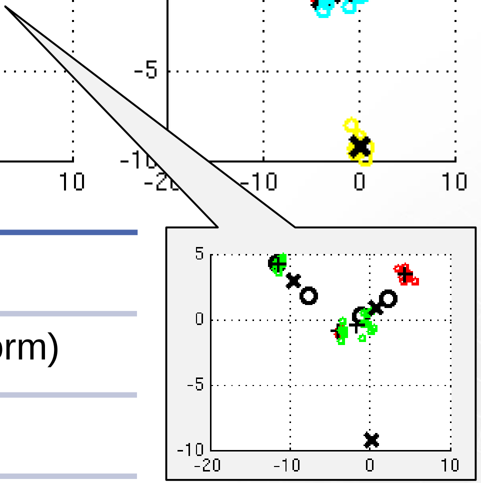
$$1 < m < \infty ,$$

X-means

- Starts from minimal number of clusters (~2)
- Every iteration every centroid is split up to 2 centroids for which k-means procedure provided
- Splitting assumed as success if BIC of two split clusters larger than BIC of a single cluster
- Stop when splitting of any centroids doesn't bring a gain in BIC



x-means	
	+ High flexibility
	- Non-stable clustering (uniform)
	- High complexity



$$\arg\min_{S,C} \sum_{j=1}^{C} \sum_{x_i \in S_j} \left\| x_i - \mu_j \right\|^2 ,$$

Splitting:

$$\mu_j^{(1,2)} = \mu_j \pm \delta , \qquad \mu_j = \frac{1}{|S_j|} \sum_{x \in S_j} x_i .$$

$$C' = \begin{cases} C, & F(\mu_j) \geq F(\mu_j^{(1)}, \mu_j^{(2)}) \\ C+1, & F(\mu_j) < F(\mu_j^{(1)}, \mu_j^{(2)}) \end{cases} ,$$

Split model scoring

Bayesian Information Criterion (BIC) as cluster location and number of clusters optimization

$$F_{BIC}(M_j) = \hat{l}_j(D) - \frac{p_j}{2} \cdot \log(R)\,,$$

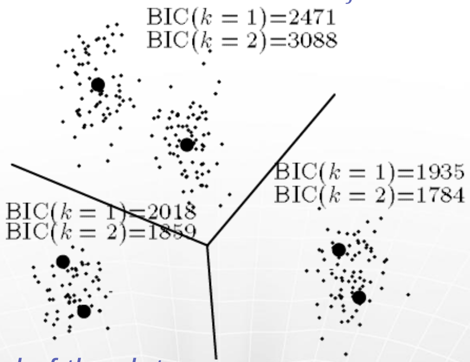
Posterior probability is used to score the models (single/split clusters) by formula mentioned in [Kass, Wasserman; 1995] based on log-likelihood of the data $\hat{l}_j(D)$.

$$\hat{\sigma}^2 = \frac{1}{R-K}\sum_i \left\|x_i - \mu_{(i)}\right\|^2,$$

Data variance

$$\hat{P}(x_i) = \frac{R_{(i)}}{R} \cdot \frac{1}{\sqrt{2\pi}\hat{\sigma}^M} \exp\left(-\frac{1}{2\hat{\sigma}^2}\left\|x_i - \mu_{(i)}\right\|^2\right),$$

Point probabilities



$$l(D) = \log \prod_i P(x_i) = \sum_i \left(\log \frac{1}{\sqrt{2\pi}\sigma^M} - \frac{1}{2\sigma^2}\left\|x_i - \mu_{(i)}\right\|^2 + \log \frac{R_{(i)}}{R}\right),$$

Log-likelihood of the data

$$\hat{l}(D_n) = -\frac{R_n}{2}\log(2\pi) - \frac{R_n \cdot M}{2}\log(\hat{\sigma}^2) - \frac{R_n - K}{2} + R_n \log R_n - R_n \log R\,.$$

Maximum likelihood estimation of the set D_n belonging to centroid μ_n

M – number of dimensions
K – number of clusters
p_j – number of free parameters

D – input set of points
$D_i \in D$ – set of points attached to μ_i centroid
$R = |D|, R_i = |D_i|$ – number of points

Simulation results. Random generated channels

BIC_SUM: Sum of clusters BIC; BIC_MEAN: Mean of clusters BIC; BIC_DEV: Deviation of clusters BIC; Clu_NUM: Num of clusters; Clu_MAX: Clusters max volume; Clu_MIN: Clusters min volume; Clu_MEAN: Clusters mean volume; Clu_DEV: Clusters volume's deviation

UE distrib.	Cluster limit		BIC_SUM	BIC_MEAN	BIC_DEV	Clu_NUM	Clu_MAX	Clu_MIN	Clu_MEAN	Clu_DEV
HIGH CORR. (Groups = 5; Deviation = 0.01)	Ideal (Cluster Limit = 5)	k-means	-3778.4	-755.67	634.21	5	18.32	4.542	10	5.3132
		c-means	-3921.5	-784.31	229.2	5	13.289	6.901	10	2.2926
		x-means	-3766.5	-753.3	0.10147	5	10	10	10	0
	Oversize (Cluster Limit = 10)	k-means	-1582.1	-158.21	407.82	10	10.722	1.217	5	3.423
		c-means	-2438.3	-243.83	355.04	10	9.996	1.049	5	3.7347
		x-means	-3766.5	-753.3	0.10079	5	10	10	10	0
UNIFORM (Groups = 5; Deviation = 1.0)	Ideal (Cluster Limit = 5)	k-means	-4069.6	-813.92	248.16	5	13.09	7.854	10	1.9821
		c-means	-4201	-840.2	465.87	5	15.133	4.314	10	4.0645
		x-means	-4057.9	-811.58	871.49	5	20.127	1	10	6.9543
	Oversize (Cluster Limit = 10)	k-means	-1867.2	-186.72	393.04	10	9.987	1.337	5	3.1251
		c-means	-2066.7	-206.67	446.69	10	10.074	0.746	5	3.7053
		x-means	-1844.3	-184.43	556.62	10	10.792	1	5	4.411

Clustering: comparative analysis

	Random generated channel				Status
Distribution	*High-correlated*		*Uniform*		
Cluster limit	*Ideal*	*Oversize*	*Ideal*	*Oversize*	
k-means	!	✗	✓	✓	*Initialization of centroids optimization required*
c-means	!	✗	✓	✓	*Initialization of centroids optimization required*
x-means	✓	✓	!	!	*1. Criterion optimization* *2. Splitting rule definition*

k-means

+ Acceptable complexity
+ Guaranteed clustering (uniform distribution)
- Low flexibility (fixed num of clusters)
- Initial step dependence

c-means

+ High after-clustering versatility
+ Guaranteed clustering (uniform distribution)
- Low flexibility (fixed num of clusters)
- Initial step dependence

x-means

+ High flexibility
+ BIC-based metric
- Non-stable clustering (uniform)
- High complexity

Robust Scheduling: propagation channel dynamics

Coherence time of channel plays an important role here. From theoretical point of view, coherence time T_c can define as

$$T_c = \frac{\lambda}{2v} = \frac{c}{2f_c \cdot v},$$

where c - speed of light, λ - wave length on central frequency f_c, v - speed of user.

According ray-tracing model of channel representation, each tap of discrete channel impulse response (τ_i) can be interpreted as response of in-depended ray(s), which passed through channel and according some distances d_i from transmitter to receiver. When interval coherence in frequency domain B_c can defined as

$$B_c = \frac{c}{|max(d_i) - min(d_i)|} \approx \frac{1}{\sigma_\tau},$$

where σ_τ is delay spread of channel impulse response.

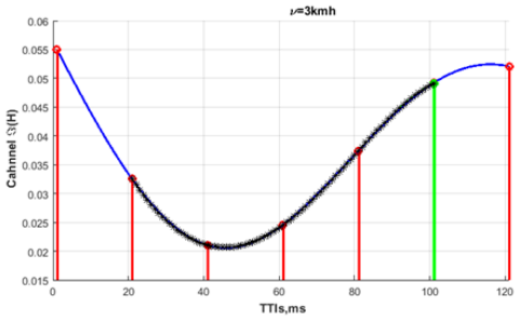
$$\tau_c = B_c T_c f.$$

Robust Scheduling: propagation channel dynamics

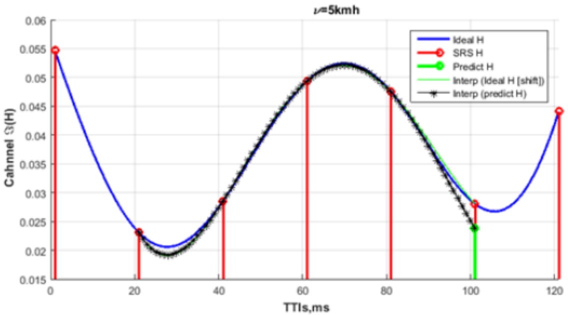
Example: $f_c = 2.6GHz$; $\sigma_\tau \approx 1.3us$; $f = 15.36MHz$.

$v = 1kmh$	$T_c \approx 208ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 10\ ms$
$v = 2kmh$	$T_c \approx 104ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 5\ ms$
$v = 3kmh$	$T_c \approx 70ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 3.5\ ms$
$v = 5kmh$	$T_c \approx 41ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 2\ ms$
$v = 7kmh$	$T_c \approx 30ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 1.5\ ms$
$v = 10kmh$	$T_c \approx 21ms, B_c \approx 770k\text{Hz} \rightarrow \tau_c \approx 1\ ms$

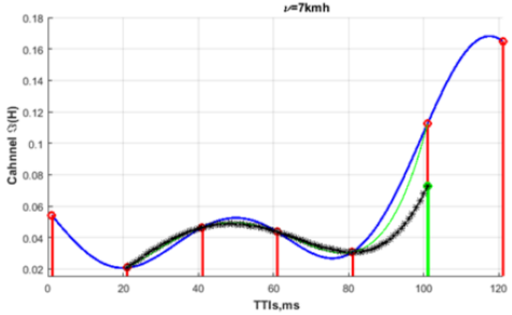
Robust Scheduling: propagation channel dynamics



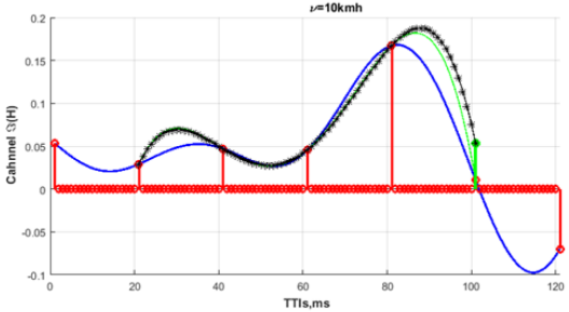
a) Channel varying in time **H**(t) with Ts=1ms (blue line), Ts=20ms (red points). Channel prediction point **H**(t+1) (green point) is estimated by AR model (channel samples inside coherence interval), $\Delta H \rightarrow 0$



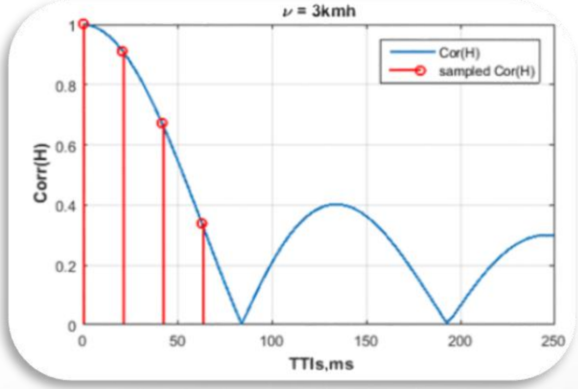
b) Channel varying in time H(t) with Ts=1ms (blue line), Ts=20ms (red points). Channel prediction point H(t+1) (green point) is estimated by AR model (channel samples partially outside coherence interval), low error of prediction $\Delta H \approx 0$

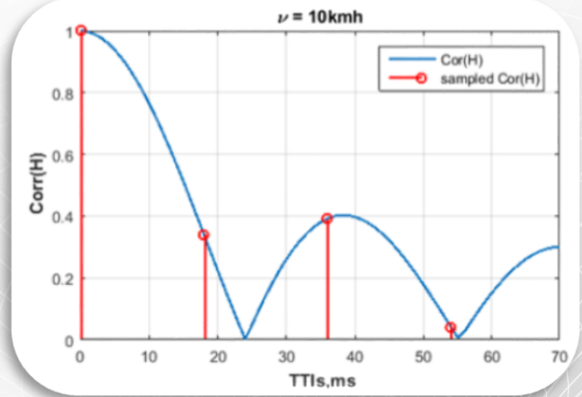


c) Channel varying in time H(t) with Ts=1ms (blue line), Ts=20ms (red points). Channel prediction point H(t+1) (green point) is estimated by AR model (more when half of channel samples outside coherence interval), error of prediction $\Delta H > 0$ is significant



d) Channel varying in time H(t) with Ts=1ms (blue line), Ts=20ms (red points). Channel prediction point H(t+1) (green point) is estimated by AR model (channel samples on the edge of aliasing), error of prediction $\Delta H > 0$ is significant





Robust scheduling: UE scores

$$Q_i = 10\log_{10}\frac{1 - C_{\{S_i\}_{i=1}^N}}{\frac{1}{SNR_i} + \alpha(t)\frac{C_{\{S_i\}_{i=1}^N}}{N_L}\Delta v_i(t)}$$

$$C_{\{S_i\}_{i=1}^N} = \sum_{\substack{j\neq i \\ j=1\ldots N}} v_{S_i}^H v_{S_j}$$

15% cell throughput gain
30% coverage area extension

$$\Delta v_i(t) = \left\| v_i(T) - \left(\tilde{v}_i^H(T)v_i(T)\right)\,\tilde{v}_i(T)\right\|^2$$

$$\alpha(t)\colon \begin{cases} \alpha(0) = 0; \\ \langle some\ function\rangle, t \in (0,T); \\ \alpha(T-\varepsilon) \approx 1. \end{cases}$$

Further questions…

- Complex channel vector clustering criterion definition
- X-means: clusters splitting mechanism optimization
- K/C-means: initialization of centroids optimization
- K/X-means: after-clustering correlation control
- Permanent partial cluster update

- How to define “the best” coupling function $\alpha(t)$

THANK YOU