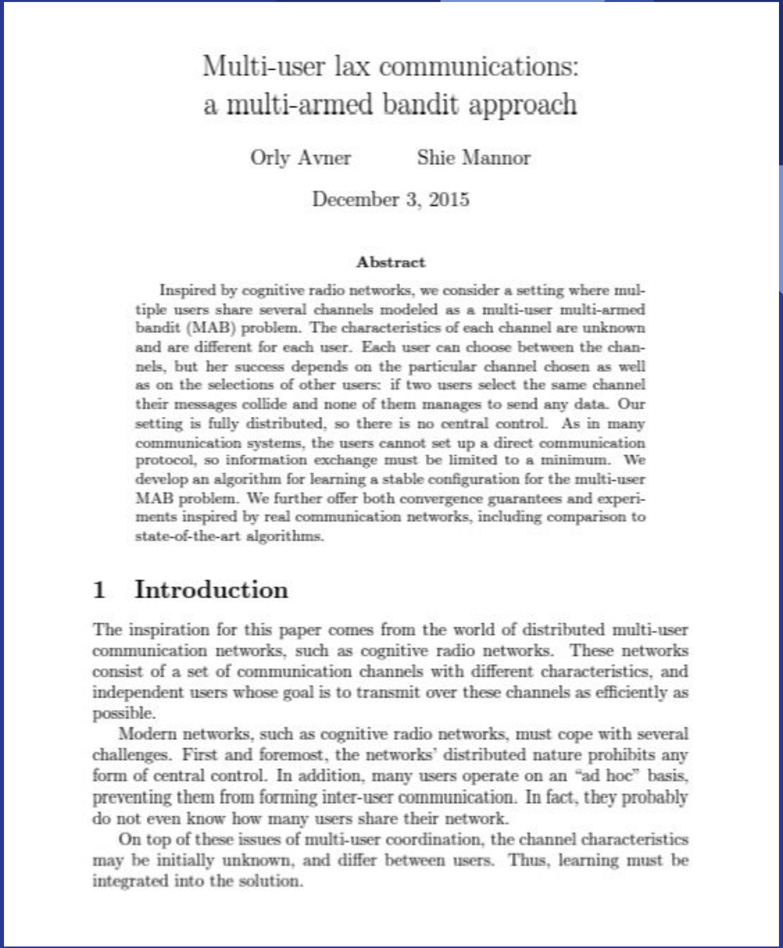
Multi-user lax communications: a multi-armed bandit approach

Report By:
Aditya (2019 BCS-002)
Sarthak Jain (2019 BCS-055)
Yuvaan Singh (2019 BCS-074)

Authors:
Orly Avner and Shie Mannor
Publications:
IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1-9, doi: 10.1109/INFOCOM.2016.7524557.



Multi-user lax communications:
a multi-armed bandit approach

Orly Avner Shie Mannor

December 3, 2015

Abstract

Inspired by cognitive radio networks, we consider a setting where multiple users share several channels modeled as a multi-user multi-armed bandit (MAB) problem. The characteristics of each channel are unknown and are different for each user. Each user can choose between the channels, but her success depends on the particular channel chosen as well as on the selections of other users: if two users select the same channel their messages collide and none of them manages to send any data. Our setting is fully distributed, so there is no central control. As in many communication systems, the users cannot set up a direct communication protocol, so information exchange must be limited to a minimum. We develop an algorithm for learning a stable configuration for the multi-user MAB problem. We further offer both convergence guarantees and experiments inspired by real communication networks, including comparison to state-of-the-art algorithms.

1 Introduction

The inspiration for this paper comes from the world of distributed multi-user communication networks, such as cognitive radio networks. These networks consist of a set of communication channels with different characteristics, and independent users whose goal is to transmit over these channels as efficiently as possible.

Modern networks, such as cognitive radio networks, must cope with several challenges. First and foremost, the networks' distributed nature prohibits any form of central control. In addition, many users operate on an "ad hoc" basis, preventing them from forming inter-user communication. In fact, they probably do not even know how many users share their network.

On top of these issues of multi-user coordination, the channel characteristics may be initially unknown, and differ between users. Thus, learning must be integrated into the solution.

Problem Statement

- Multiple users share several channels modeled as a multi-user multi-armed bandit (MAB) problem
- The characteristics of each channel are unknown and are different for each user.
- User doesn’t know about presence of other users in network.
- In communication systems, the users cannot set up a direct communication protocol, so information exchange must be limited to a minimum

Objective

Develop an algorithm for learning a stable configuration for the multi-user MAB problem

Assumptions

- Number of Channels(K) is greater than number of Users (N)

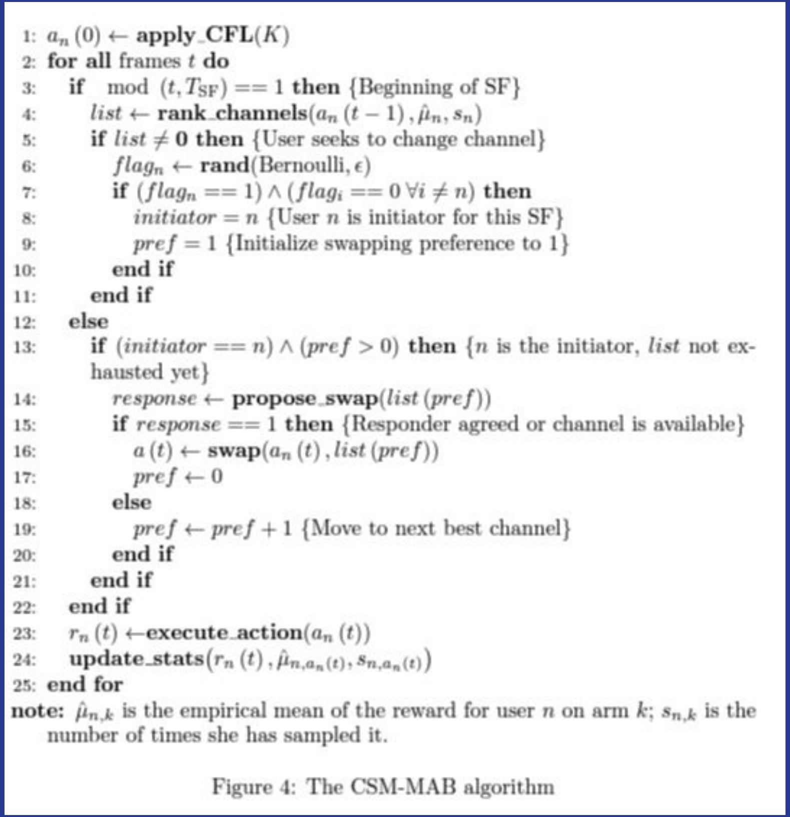
  K>=N
- Transmission failure only depends on collision of users irrespective of any transmission noise.
- We assume value of epsilon (probability to raise a flag to become initiator) as 0.2
- It is assumed that if there is collision in transmission, it results in 100% loss of communication.
- CFL (Communication Free Learning) execution stop after first orthogonalization of user-channel configuration.

Related Work

- **Hungarian Method**: Required full knowledge of the graph i.e., channel characteristics and assumes of existence central network.
- **Bertsekas auction algorithm**: Frees us from the need for central control, at the cost of direct communication between nodes.
- **Gale-Shapley algorithm**: Solves the problem of finding a stable marriage configuration, but does not take the need to learn into account.

All the above works have problems like the communication is more and their formation does not consider the learning.

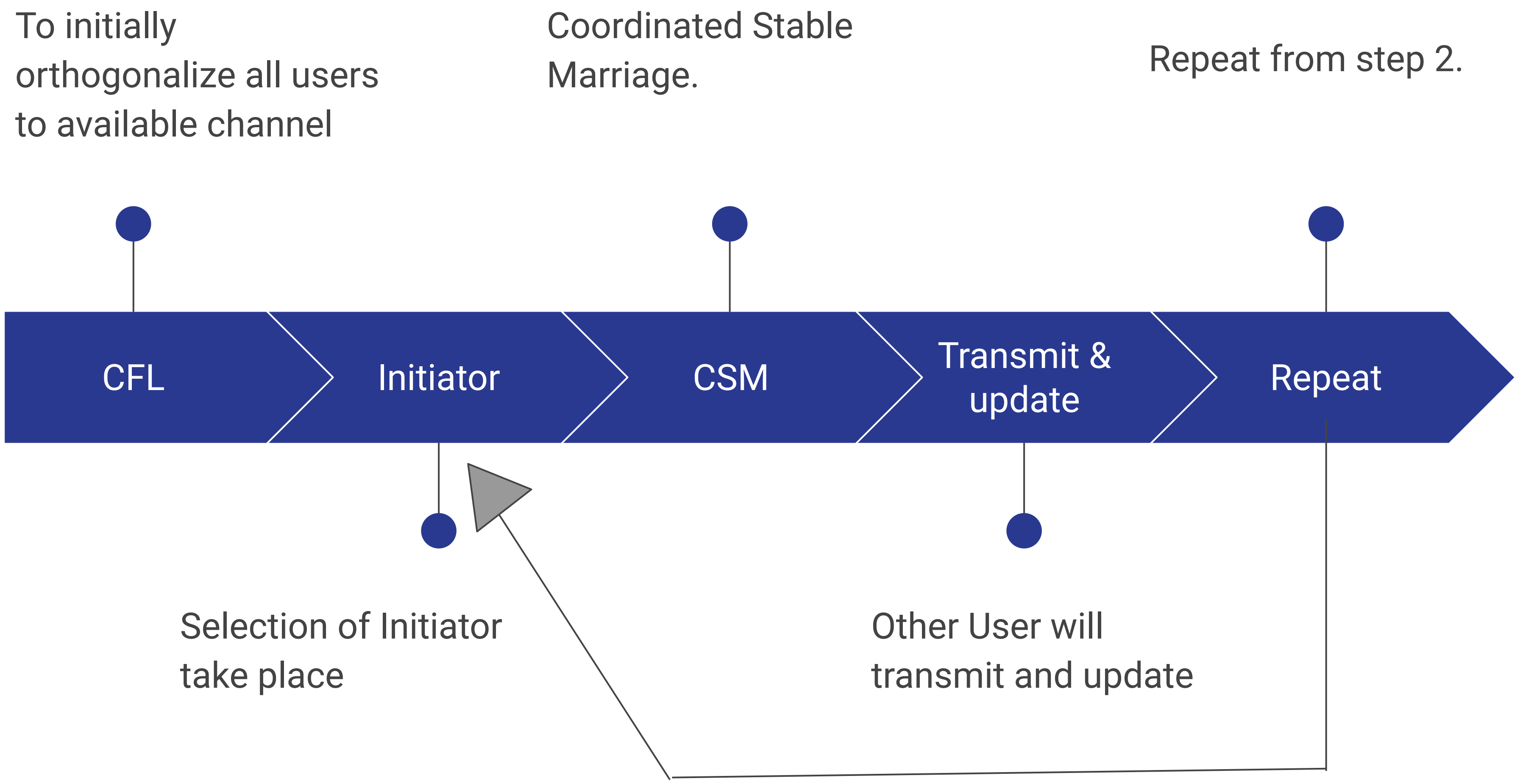
Algorithm Flow



```
1:  $a_n(0) \leftarrow$ **apply_CFL**$(K)$
2:  **for all** frames $t$ **do**
3:    **if** mod $(t, T_{SF}) == 1$ **then** {Beginning of SF}
4:      $list \leftarrow$ **rank_channels**$(a_n(t-1), \hat{\mu}_n, s_n)$
5:      **if** $list \neq \mathbf{0}$ **then** {User seeks to change channel}
6:        $flag_n \leftarrow$ **rand**(Bernoulli, $\epsilon$)
7:        **if** $(flag_n == 1) \wedge (flag_i == 0\ \forall i \neq n)$ **then**
8:          $initiator = n$ {User $n$ is initiator for this SF}
9:          $pref = 1$ {Initialize swapping preference to 1}
10:       **end if**
11:     **end if**
12:   **else**
13:     **if** $(initiator == n) \wedge (pref > 0)$ **then** {$n$ is the initiator, $list$ not ex-
        hausted yet}
14:       $response \leftarrow$ **propose_swap**$(list(pref))$
15:       **if** $response == 1$ **then** {Responder agreed or channel is available}
16:         $a(t) \leftarrow$ **swap**$(a_n(t), list(pref))$
17:         $pref \leftarrow 0$
18:       **else**
19:         $pref \leftarrow pref + 1$ {Move to next best channel}
20:       **end if**
21:     **end if**
22:   **end if**
23:   $r_n(t) \leftarrow$**execute_action**$(a_n(t))$
24:   **update_stats**$(r_n(t), \hat{\mu}_{n,a_n(t)}, s_{n,a_n(t)})$
25: **end for**
```

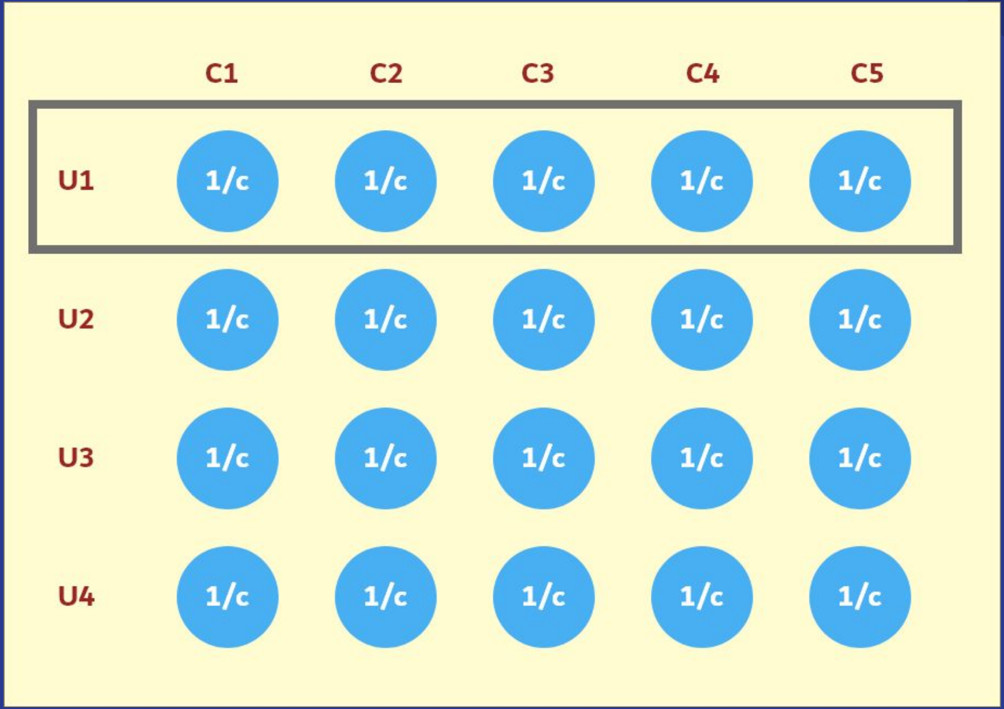
note: $\hat{\mu}_{n,k}$ is the empirical mean of the reward for user n on arm k; $s_{n,k}$ is the number of times she has sampled it.

Figure 4: The CSM-MAB algorithm


To initially orthogonalize all users to available channel
Coordinated Stable Marriage.
Repeat from step 2.
CFL
Initiator
CSM
Transmit & update
Repeat
Selection of Initiator take place
Other User will transmit and update

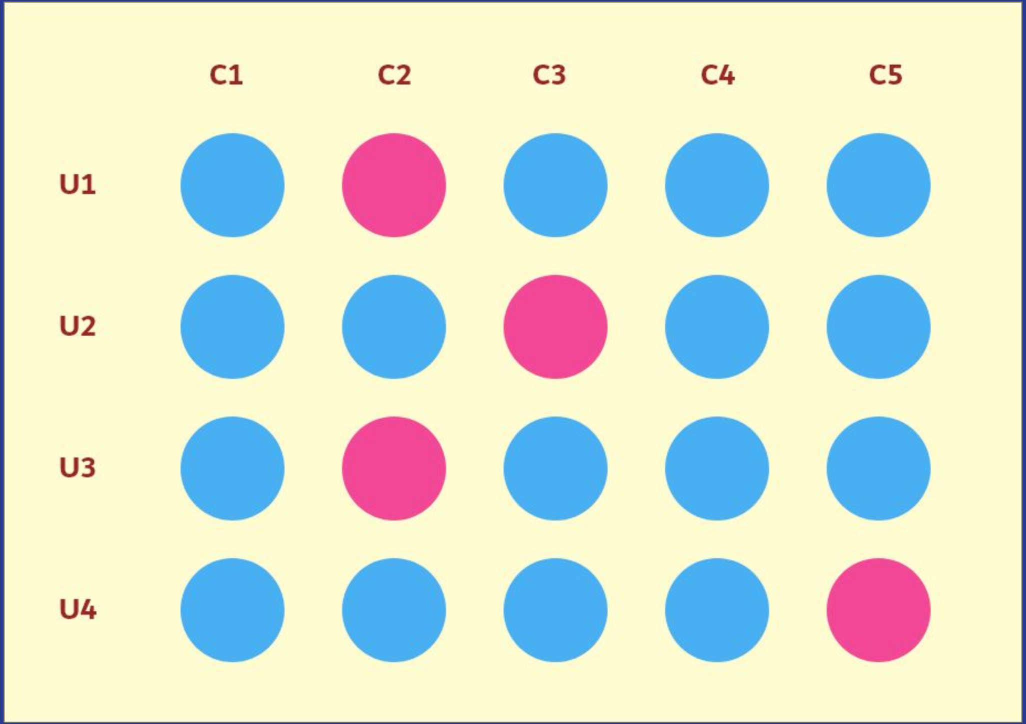
Implementation

Communication Free Learning - Implementation



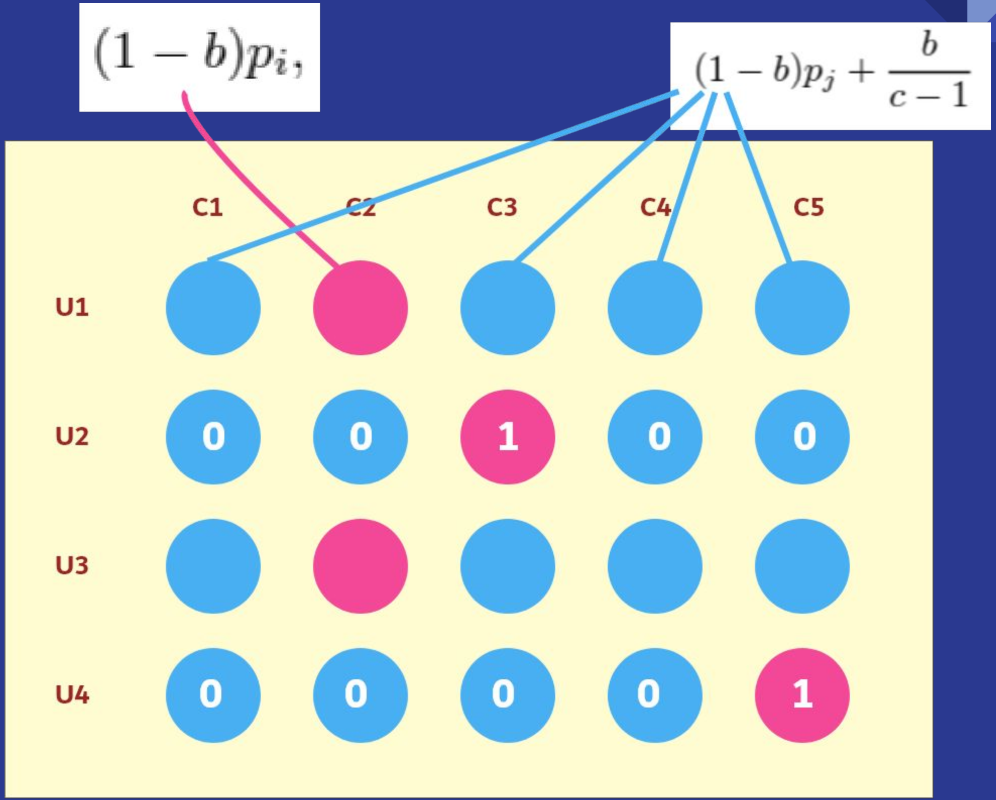
Each channel have same chance of getting selected for transmission

CFL - Implementation



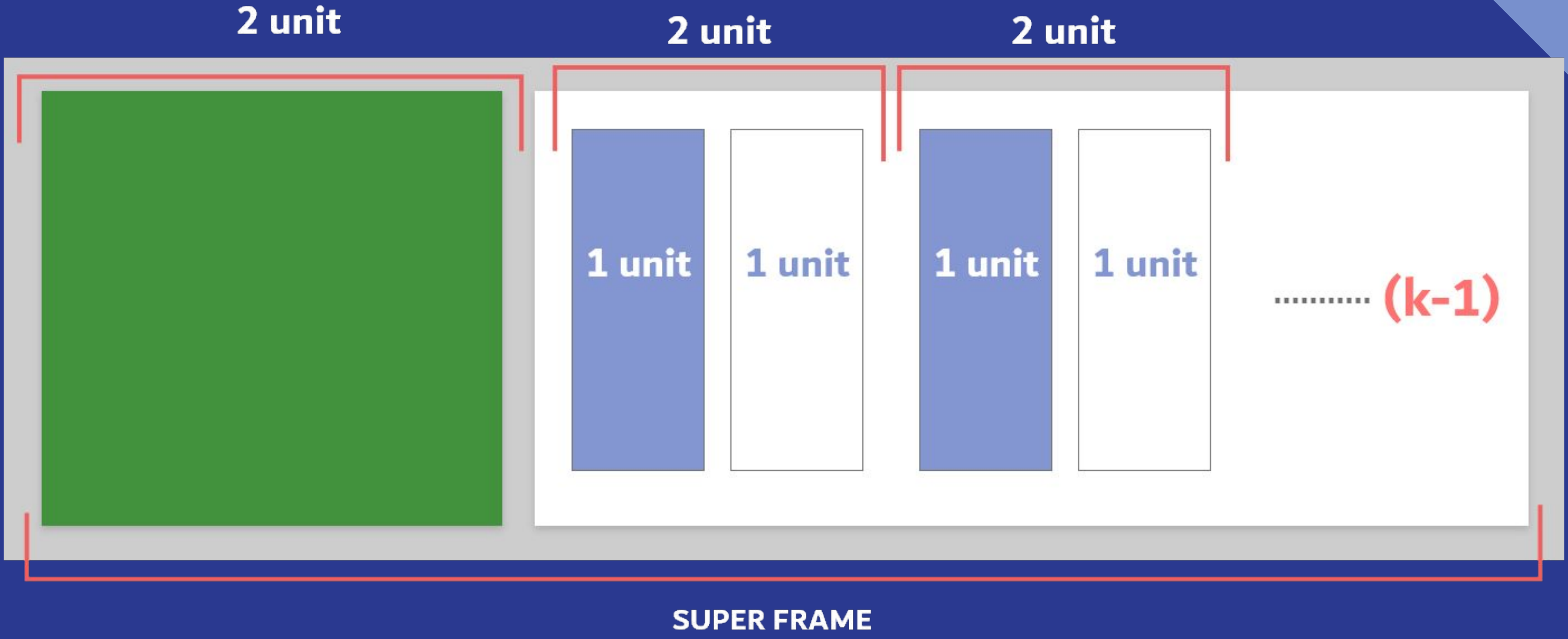
User transmitting to selected channel

CFL - Implementation



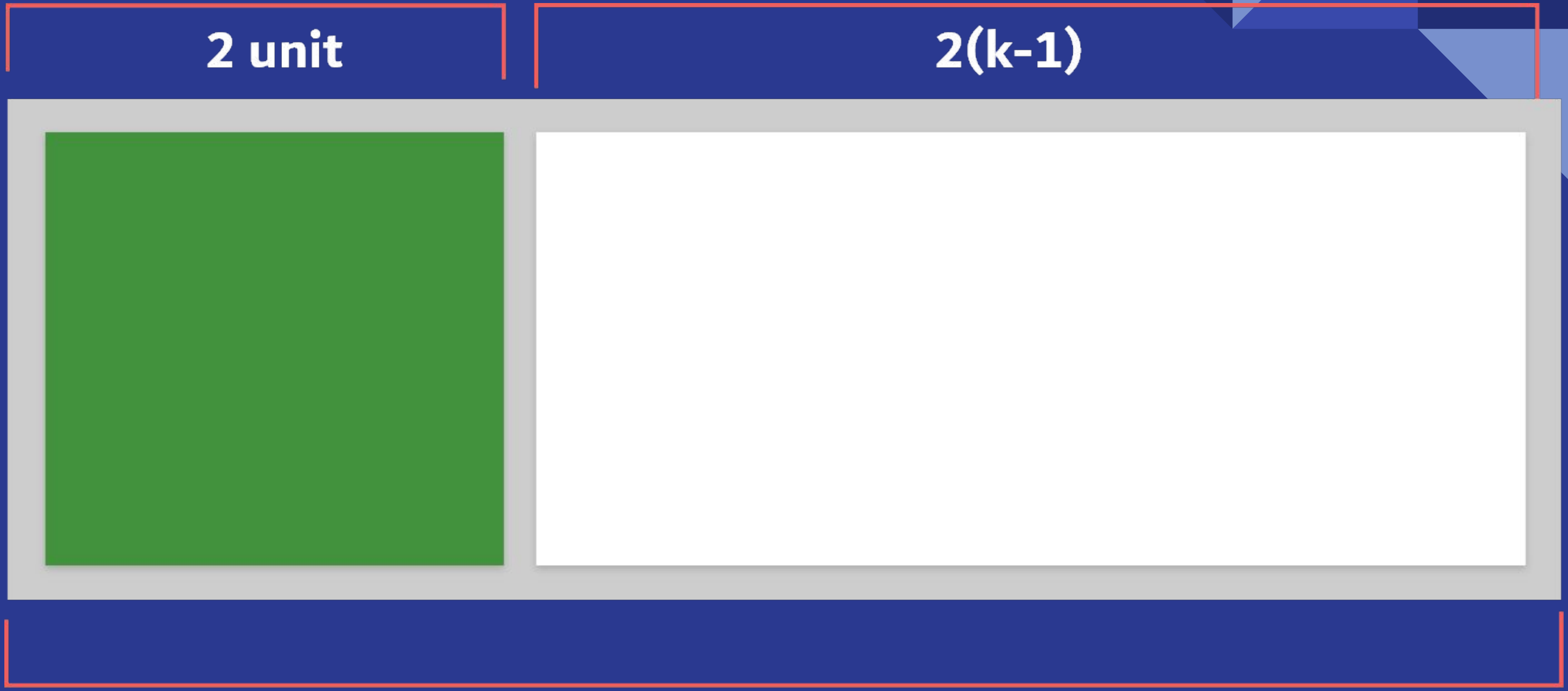
Probability update according to transmission

Super Frame Breakdown



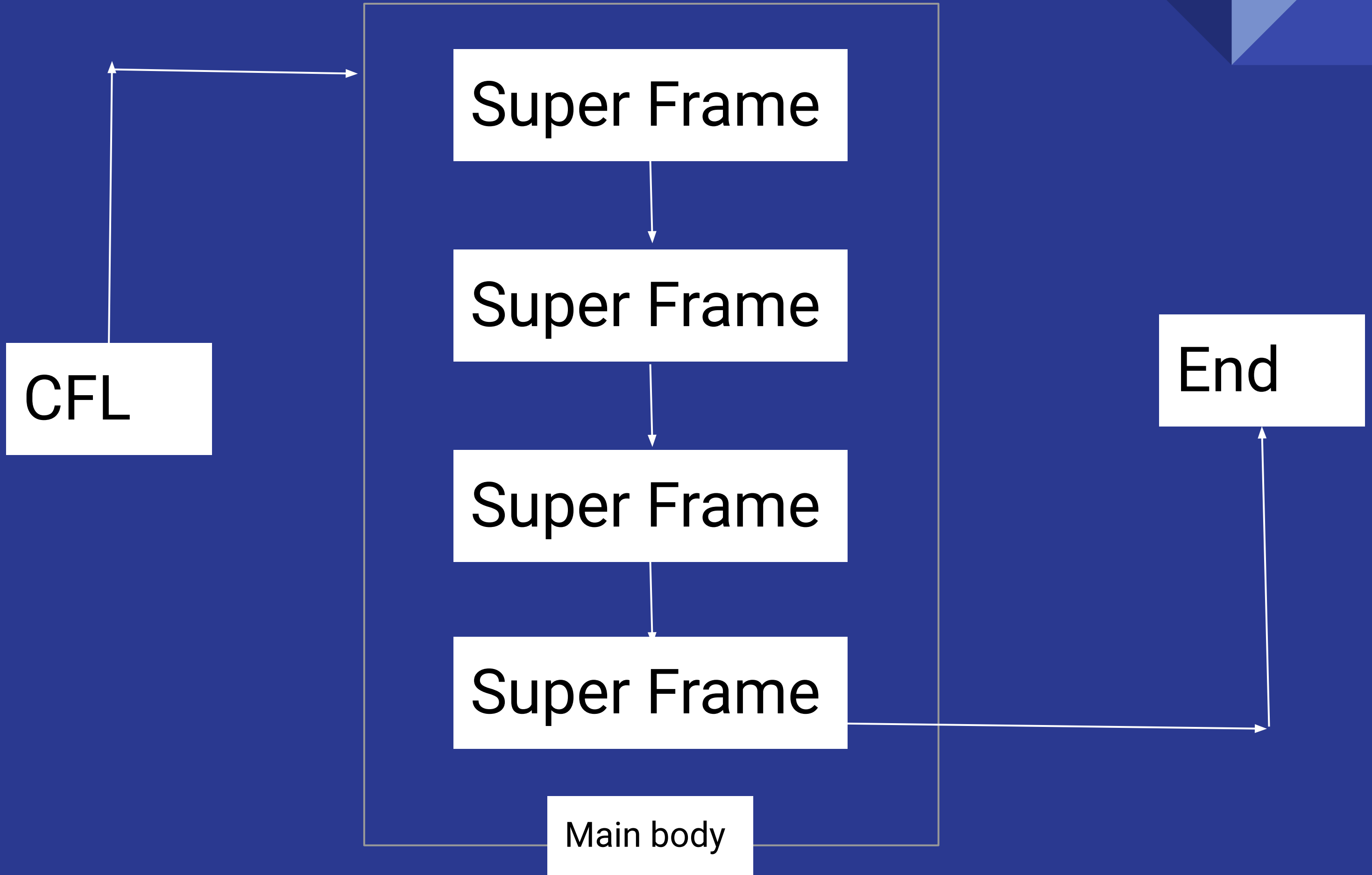
K is number of channels

Super Frame Breakdown

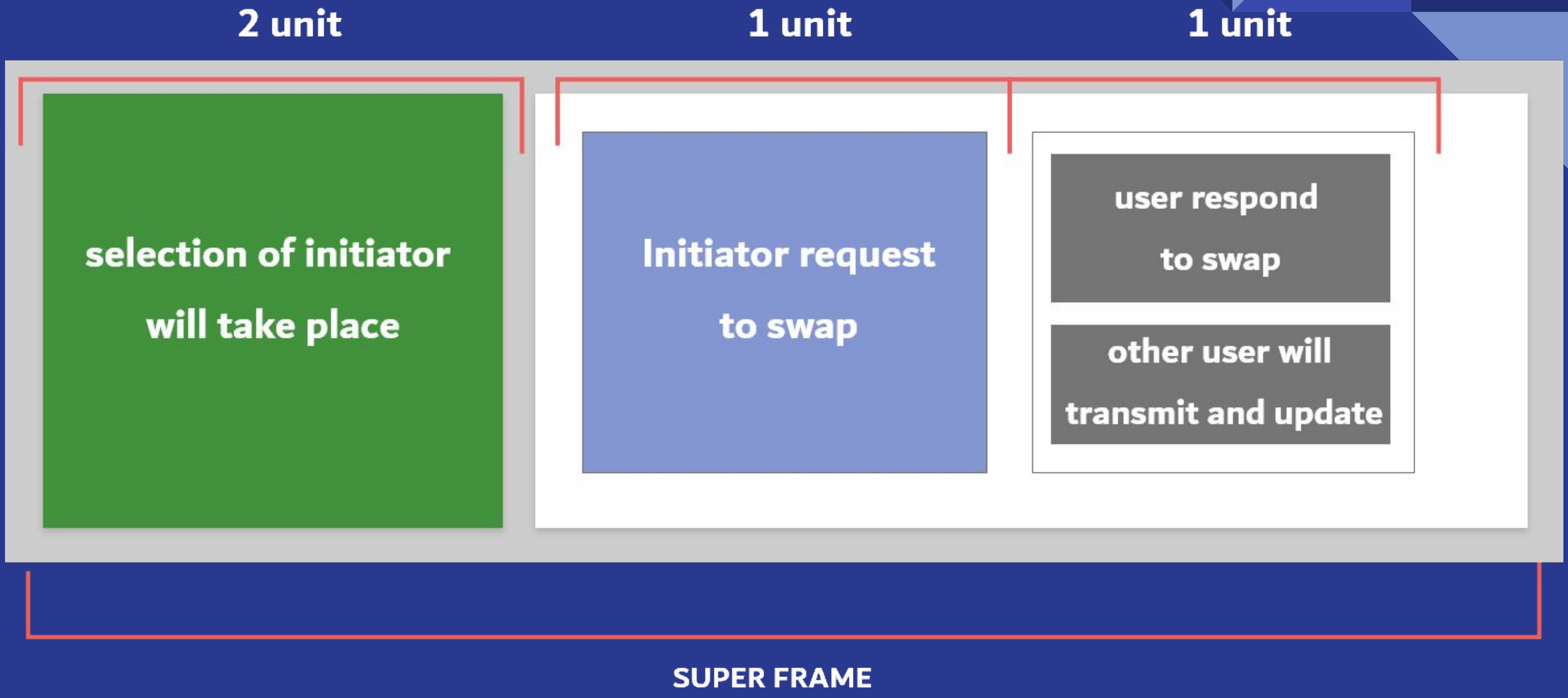


SUPER FRAME = 2 + 2(k-1)

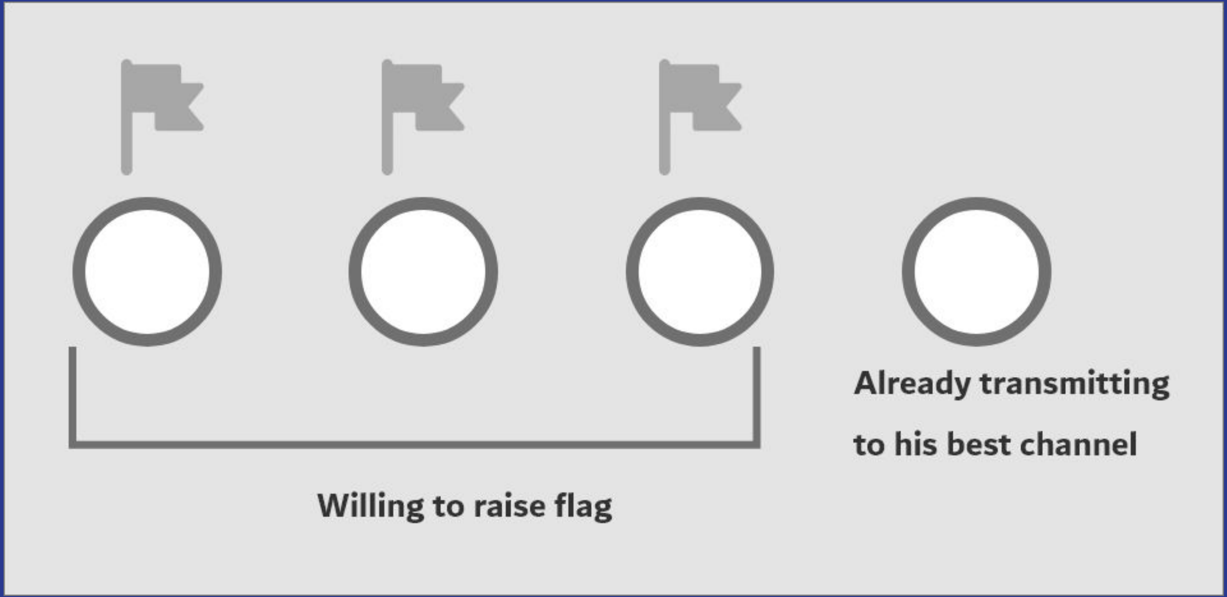
Overall Algorithm Flow



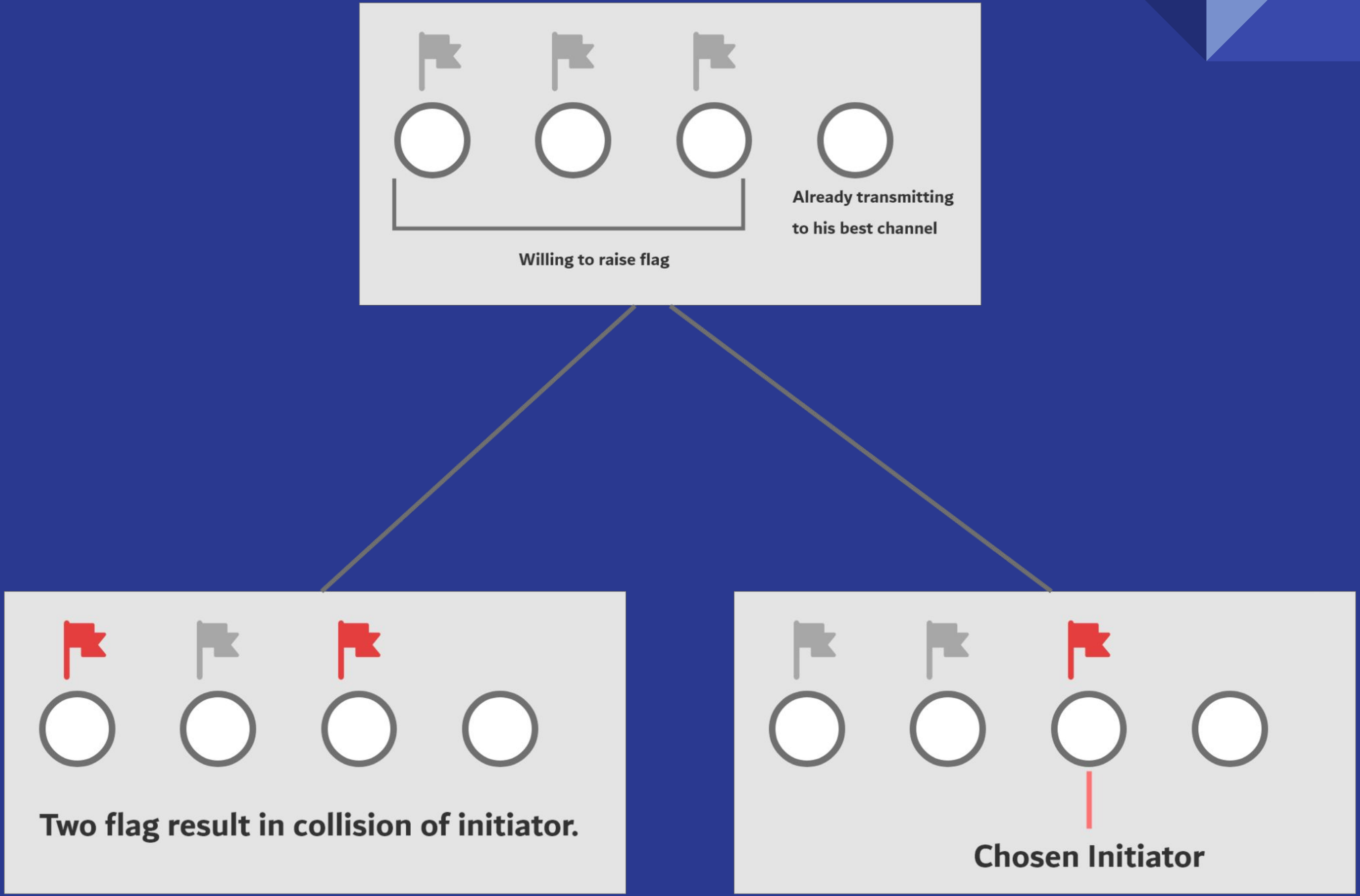
Super Frame Breakdown


2 unit
1 unit
1 unit
selection of initiator will take place
Initiator request to swap
user respond to swap
other user will transmit and update
SUPER FRAME

Initiator Selection



Initiator Selection

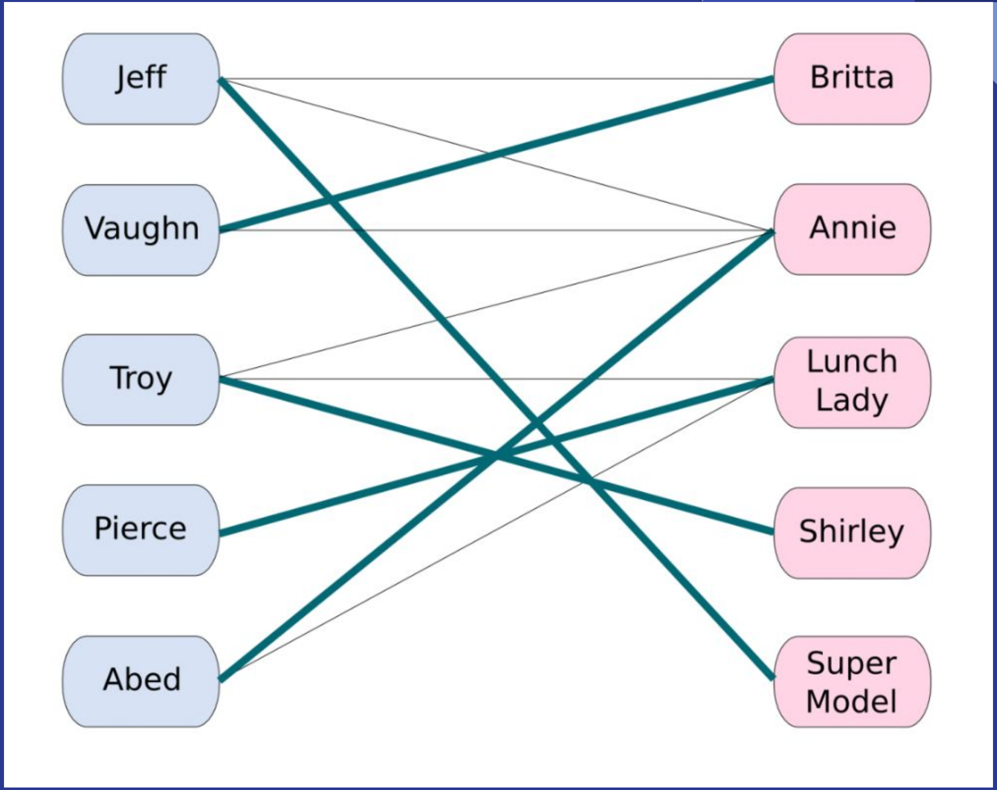

Already transmitting to his best channel
Willing to raise flag
Two flag result in collision of initiator.
Chosen Initiator

Coordinated Stable Marriage

Gale-Shapley Algo

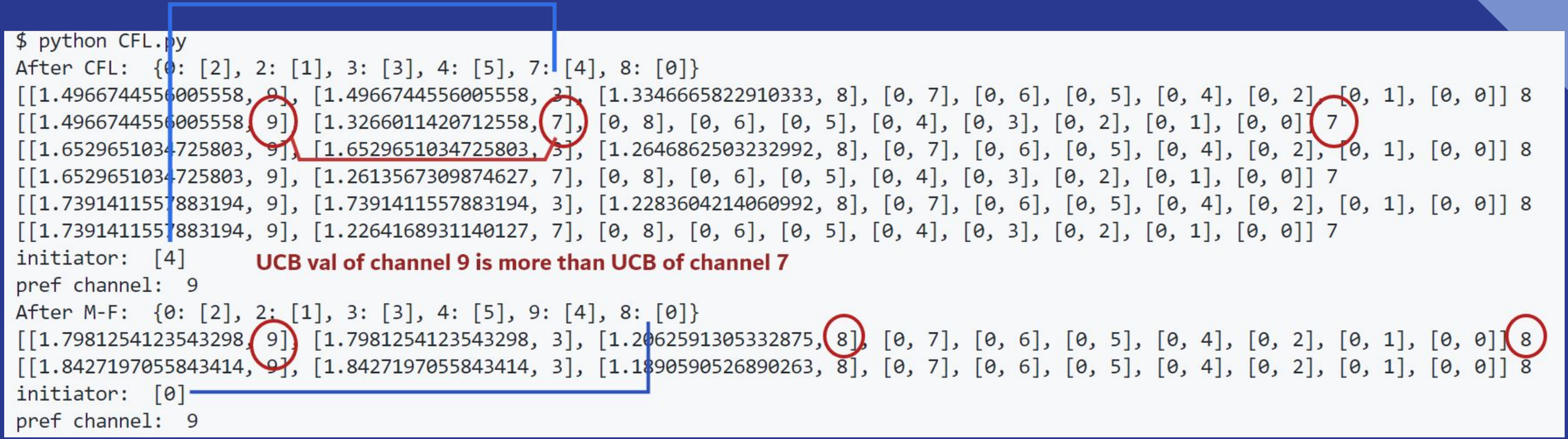
Not Optimal

Highly Efficient



Coordinated Stable Marriage

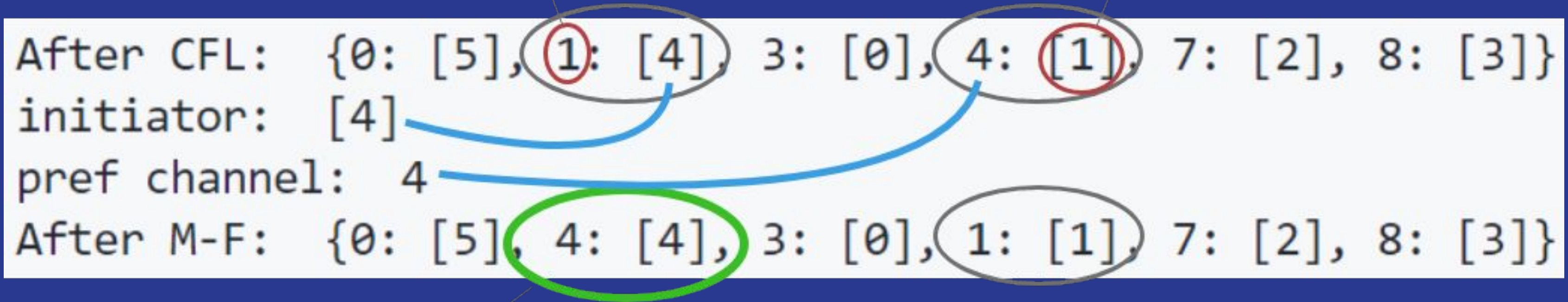
Initiator is at 7th channel



```
$ python CFL.py
After CFL:  {0: [2], 2: [1], 3: [3], 4: [5], 7: [4], 8: [0]}
[[1.4966744556005558, 9], [1.4966744556005558, 3], [1.3346665822910333, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 2], [0, 1], [0, 0]] 8
[[1.4966744556005558, 9], [1.3266011420712558, 7], [0, 8], [0, 6], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1], [0, 0]] 7
[[1.6529651034725803, 9], [1.6529651034725803, 3], [1.2646862503232992, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 2], [0, 1], [0, 0]] 8
[[1.6529651034725803, 9], [1.2613567309874627, 7], [0, 8], [0, 6], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1], [0, 0]] 7
[[1.7391411557883194, 9], [1.7391411557883194, 3], [1.2283604214060992, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 2], [0, 1], [0, 0]] 8
[[1.7391411557883194, 9], [1.2264168931140127, 7], [0, 8], [0, 6], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1], [0, 0]] 7
initiator:  [4]
pref channel:  9
After M-F:  {0: [2], 2: [1], 3: [3], 4: [5], 9: [4], 8: [0]}
[[1.7981254123543298, 9], [1.7981254123543298, 3], [1.2062591305332875, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 2], [0, 1], [0, 0]] 8
[[1.8427197055843414, 9], [1.8427197055843414, 3], [1.1890590526890263, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 2], [0, 1], [0, 0]] 8
initiator:  [0]
pref channel:  9
```

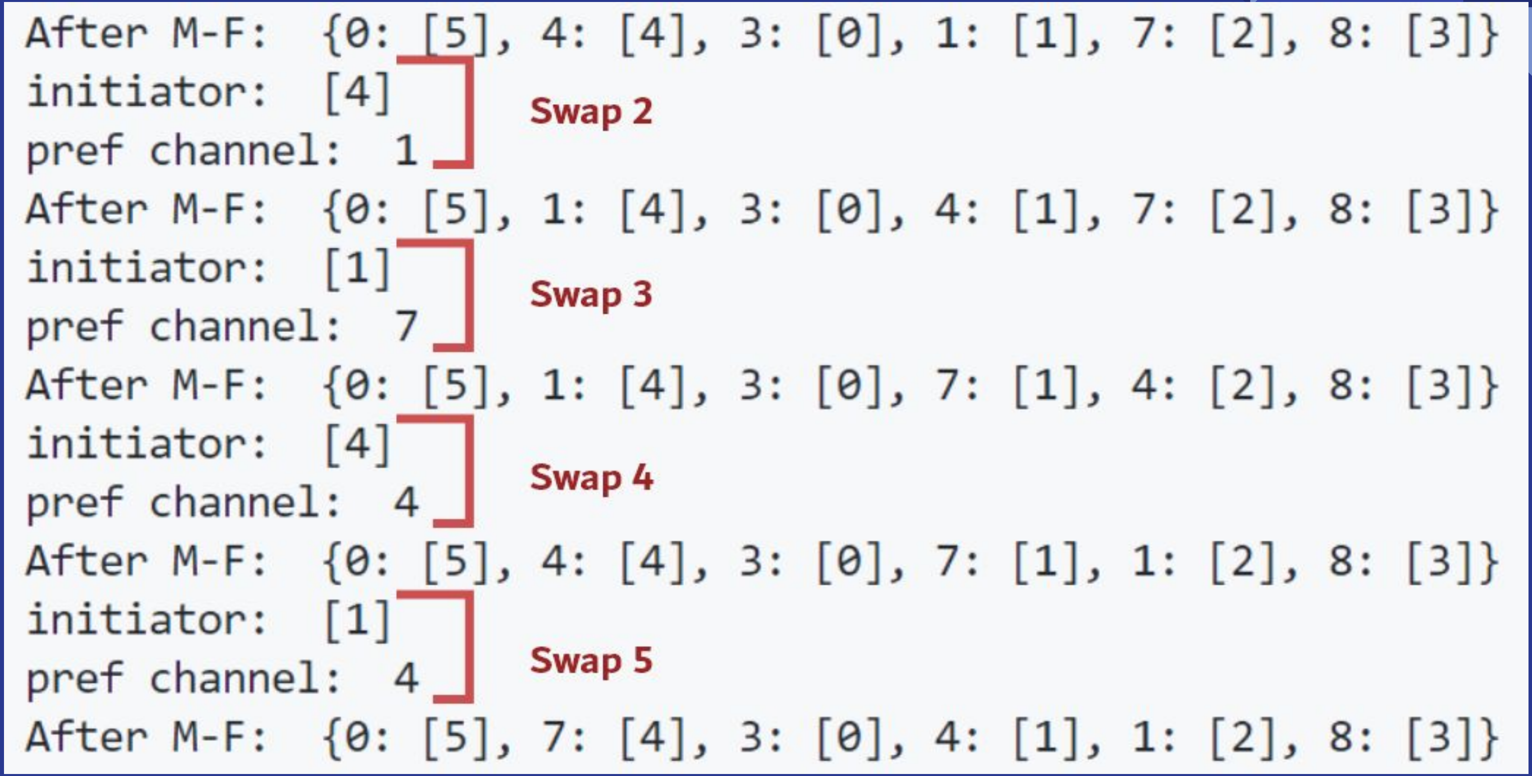
UCB val of channel 9 is more than UCB of channel 7

Coordinated Stable Marriage


Initiator is at channel 1
User 1 is at preference channel 1
Initiator at it's preference channel

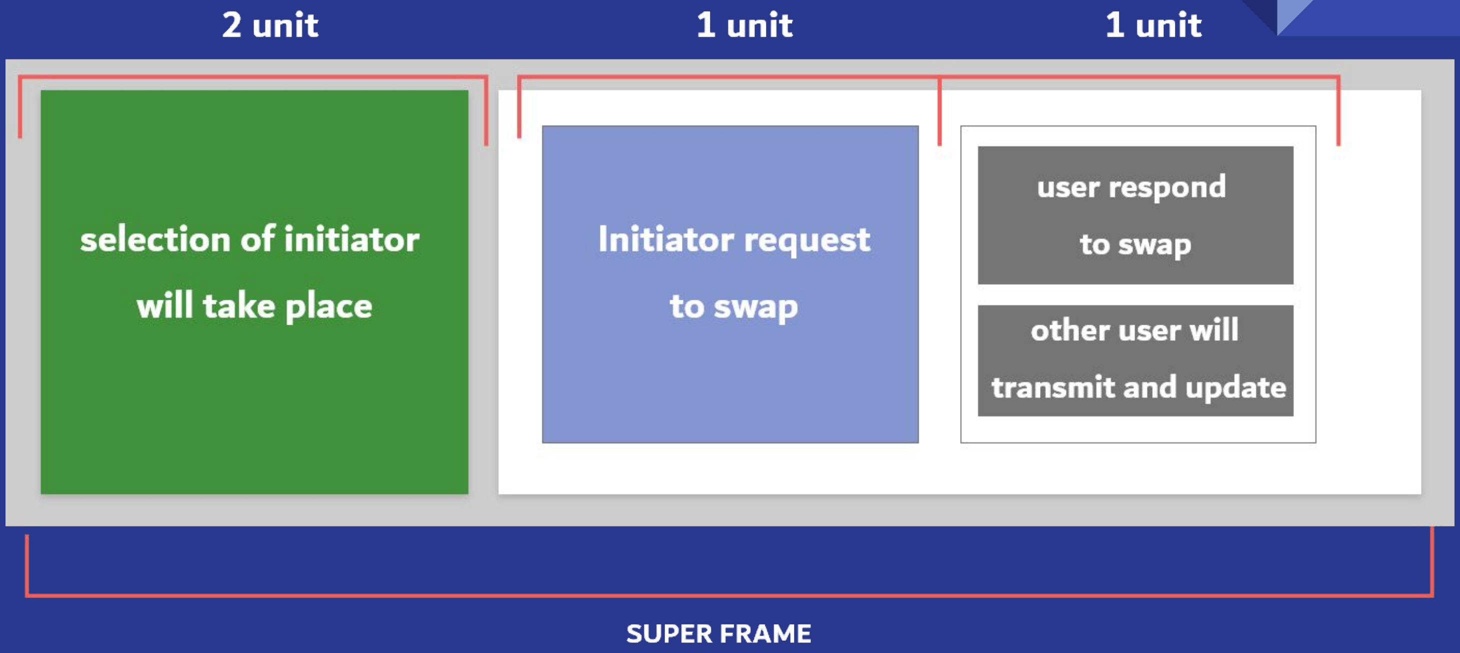
```
After CFL:   {0: [5], 1: [4], 3: [0], 4: [1], 7: [2], 8: [3]}
initiator:   [4]
pref channel:  4
After M-F:   {0: [5], 4: [4], 3: [0], 1: [1], 7: [2], 8: [3]}
```

Coordinated Stable Marriage



```
After M-F:  {0: [5], 4: [4], 3: [0], 1: [1], 7: [2], 8: [3]}
initiator:   [4]
pref channel:  1
After M-F:  {0: [5], 1: [4], 3: [0], 4: [1], 7: [2], 8: [3]}
initiator:   [1]
pref channel:  7
After M-F:  {0: [5], 1: [4], 3: [0], 7: [1], 4: [2], 8: [3]}
initiator:   [4]
pref channel:  4
After M-F:  {0: [5], 4: [4], 3: [0], 7: [1], 1: [2], 8: [3]}
initiator:   [1]
pref channel:  4
After M-F:  {0: [5], 7: [4], 3: [0], 4: [1], 1: [2], 8: [3]}
```

Single Super Frame execution complete


2 unit
1 unit
1 unit
selection of initiator will take place
Initiator request to swap
user respond to swap
other user will transmit and update
SUPER FRAME

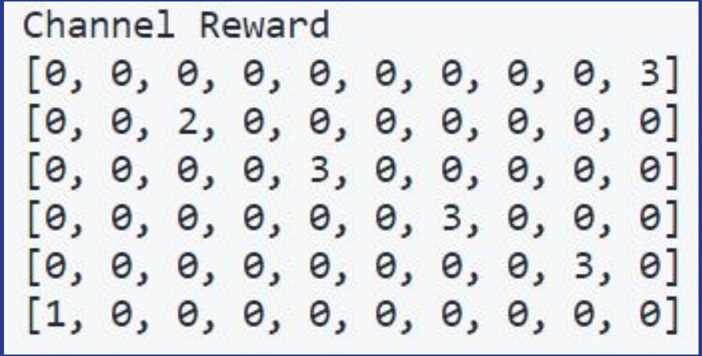
This cycle will again take place for next super frame.

Results

Channel Reward

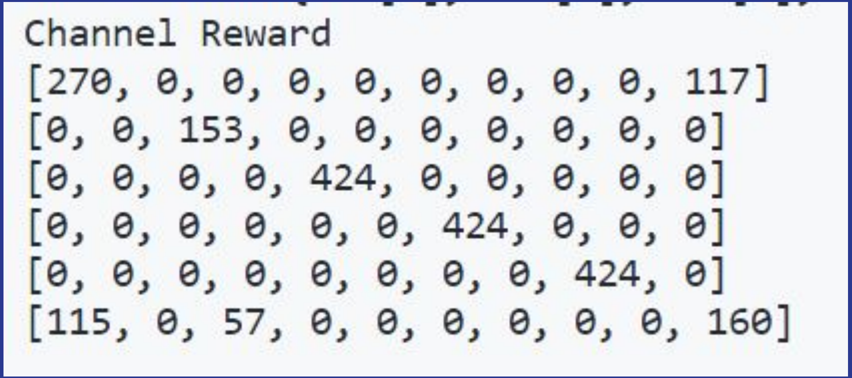
Reward is proportional to successful transmission of user to corresponding channel with associated probability of bernoulli distribution of channel selection.

Reward = bernoulli.rvs (channel selection prob)



```
Channel Reward
[0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
[0, 0, 2, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 3, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 3, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 3, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

After CFL, Before Super Frame execution

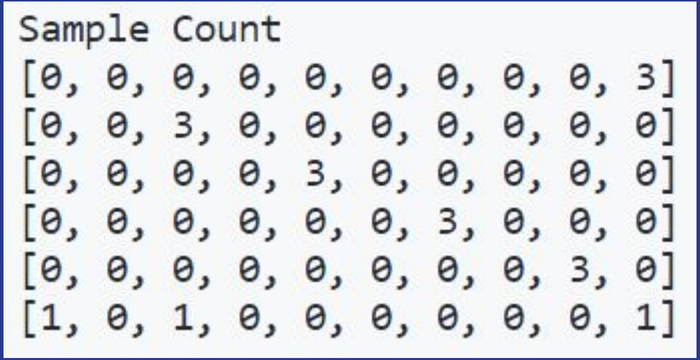


```
Channel Reward
[270, 0, 0, 0, 0, 0, 0, 0, 0, 117]
[0, 0, 153, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 424, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 424, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 424, 0]
[115, 0, 57, 0, 0, 0, 0, 0, 0, 160]
```

After Super Frame execution

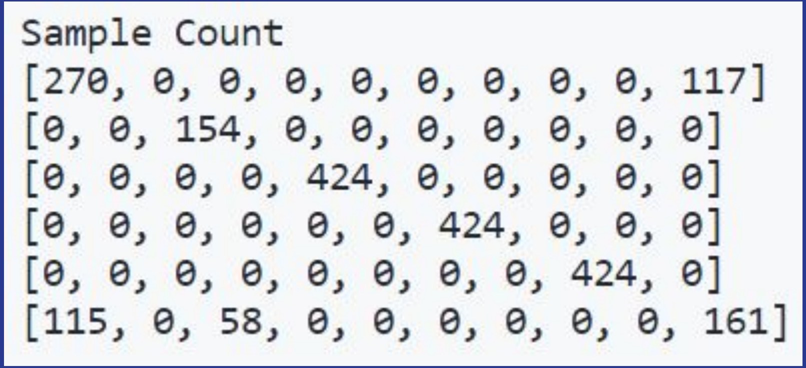
Channel Sampling

Number of times transmission is made on particular channel by a particular user.



```
Sample Count
[0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
[0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 3, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 3, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 3, 0]
[1, 0, 1, 0, 0, 0, 0, 0, 0, 1]
```

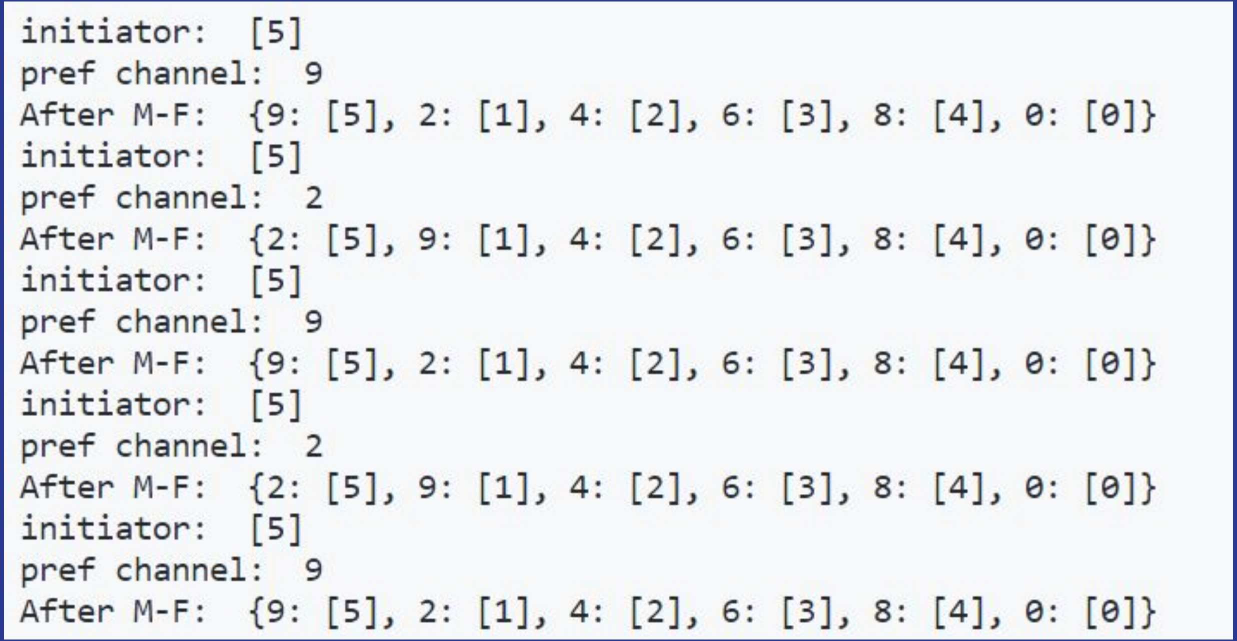
After CFL



```
Sample Count
[270, 0, 0, 0, 0, 0, 0, 0, 0, 117]
[0, 0, 154, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 424, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 424, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 424, 0]
[115, 0, 58, 0, 0, 0, 0, 0, 0, 161]
```

After Super Frame execution

Stable Marriage Swapping



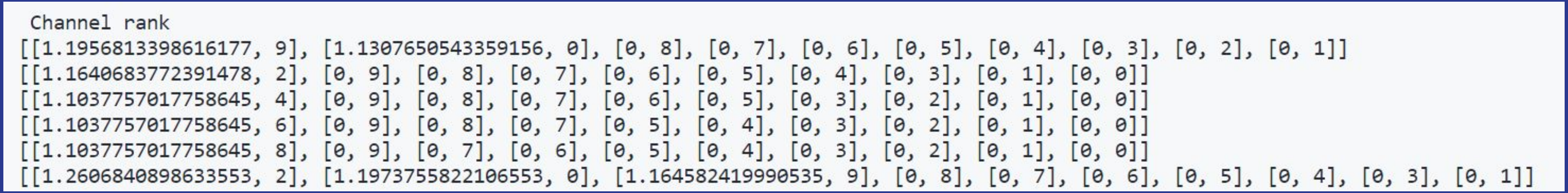
```
initiator:  [5]
pref channel:  9
After M-F:  {9: [5], 2: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
initiator:  [5]
pref channel:  2
After M-F:  {2: [5], 9: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
initiator:  [5]
pref channel:  9
After M-F:  {9: [5], 2: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
initiator:  [5]
pref channel:  2
After M-F:  {2: [5], 9: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
initiator:  [5]
pref channel:  9
After M-F:  {9: [5], 2: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
```

Channel configuration after each possible swapping in a Mini - Frame

Channel ranking

$$I_{n,k}(t) = \hat{\mu}_{n,k} + \sqrt{\frac{2 \ln t}{s_{n,k}}},$$

UCB indexing

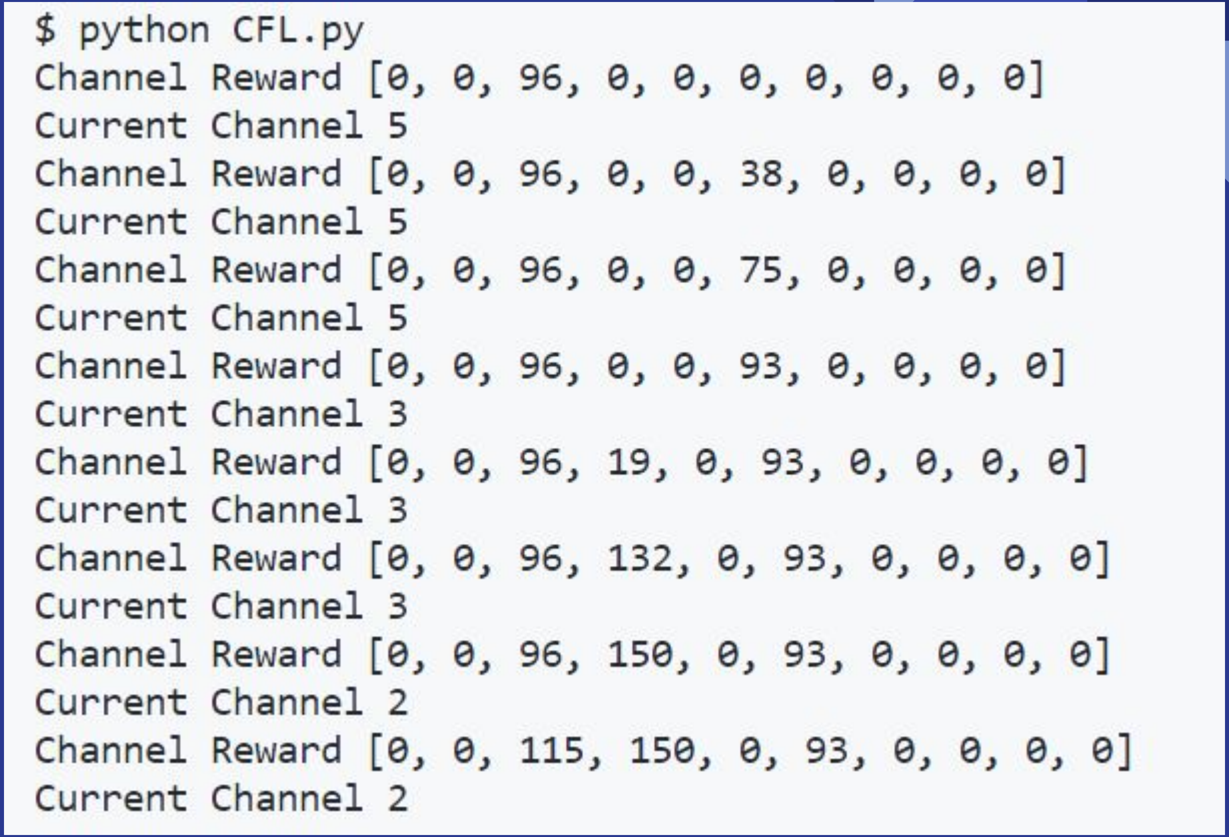


```
 Channel rank
[[1.1956813398616177, 9], [1.1307650543359156, 0], [0, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1]]
[[1.1640683772391478, 2], [0, 9], [0, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 3], [0, 1], [0, 0]]
[[1.1037757017758645, 4], [0, 9], [0, 8], [0, 7], [0, 6], [0, 5], [0, 3], [0, 2], [0, 1], [0, 0]]
[[1.1037757017758645, 6], [0, 9], [0, 8], [0, 7], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1], [0, 0]]
[[1.1037757017758645, 8], [0, 9], [0, 7], [0, 6], [0, 5], [0, 4], [0, 3], [0, 2], [0, 1], [0, 0]]
[[1.2606840898633553, 2], [1.1973755822106553, 0], [1.164582419990535, 9], [0, 8], [0, 7], [0, 6], [0, 5], [0, 4], [0, 3], [0, 1]]
```

Channel ranking based on UCB indexing

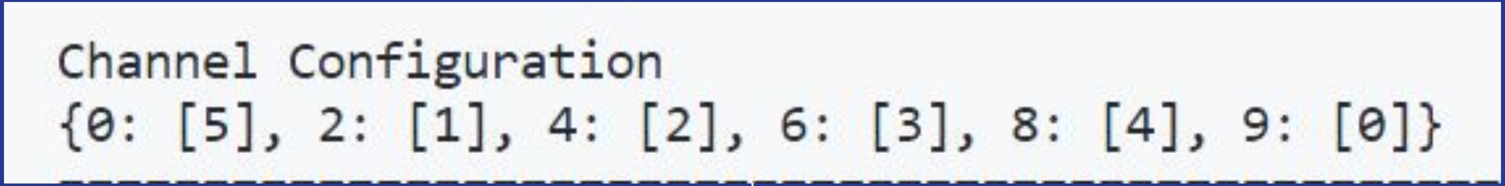
Channel reward and User swapping

Channel swapping for particular user

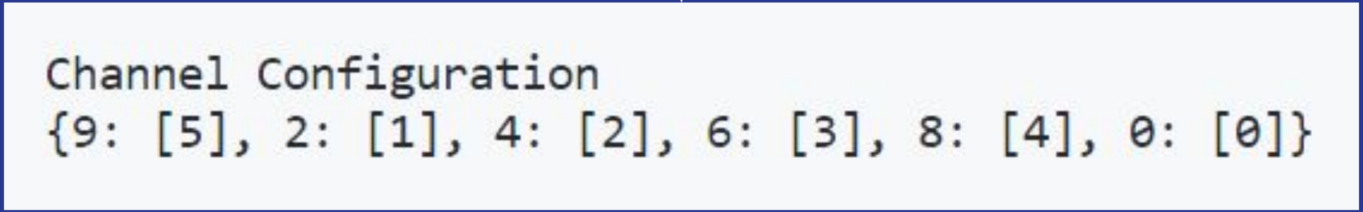


```
$ python CFL.py
Channel Reward [0, 0, 96, 0, 0, 0, 0, 0, 0, 0]
Current Channel 5
Channel Reward [0, 0, 96, 0, 0, 38, 0, 0, 0, 0]
Current Channel 5
Channel Reward [0, 0, 96, 0, 0, 75, 0, 0, 0, 0]
Current Channel 5
Channel Reward [0, 0, 96, 0, 0, 93, 0, 0, 0, 0]
Current Channel 3
Channel Reward [0, 0, 96, 19, 0, 93, 0, 0, 0, 0]
Current Channel 3
Channel Reward [0, 0, 96, 132, 0, 93, 0, 0, 0, 0]
Current Channel 3
Channel Reward [0, 0, 96, 150, 0, 93, 0, 0, 0, 0]
Current Channel 2
Channel Reward [0, 0, 115, 150, 0, 93, 0, 0, 0, 0]
Current Channel 2
```

Channel Configuration



```
Channel Configuration
{0: [5], 2: [1], 4: [2], 6: [3], 8: [4], 9: [0]}
```



```
Channel Configuration
{9: [5], 2: [1], 4: [2], 6: [3], 8: [4], 0: [0]}
```

Thank You