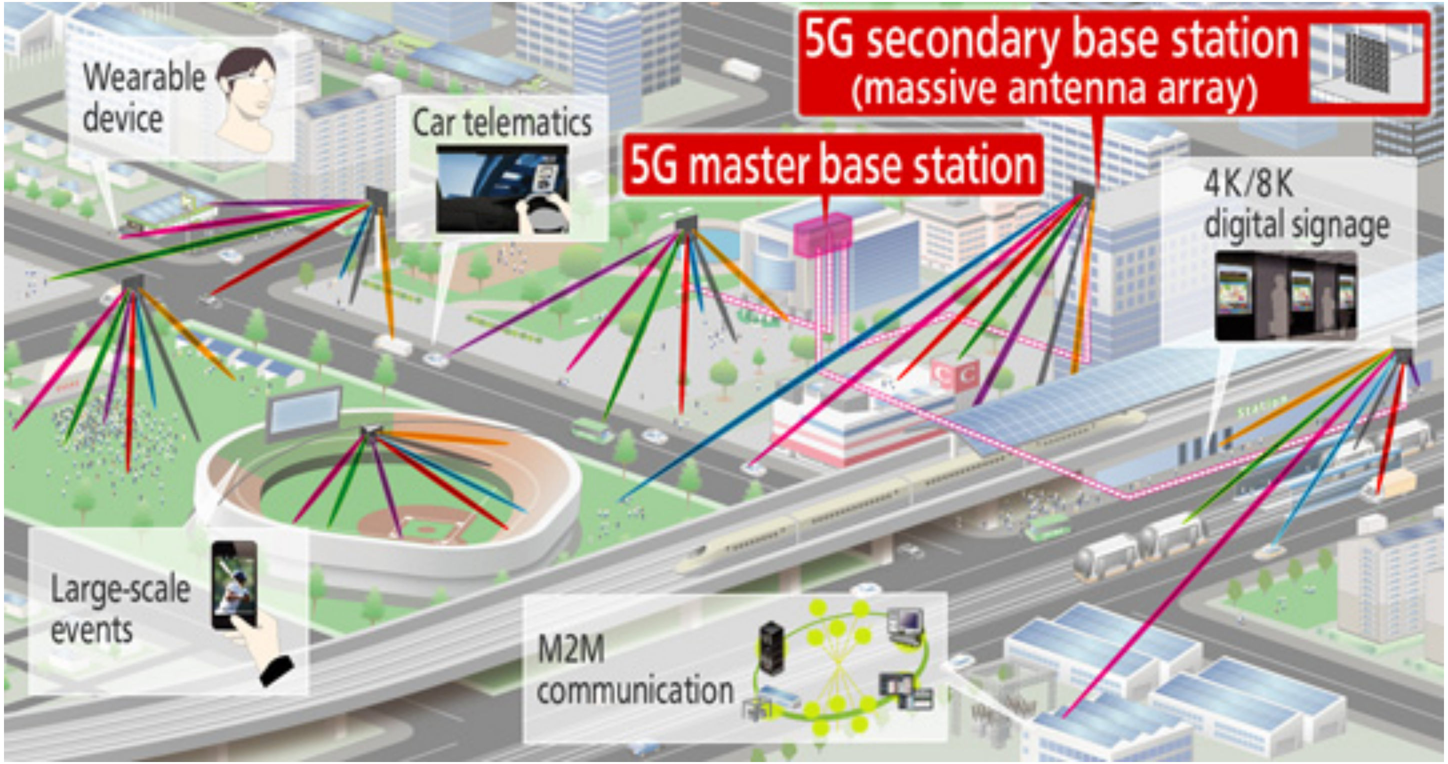
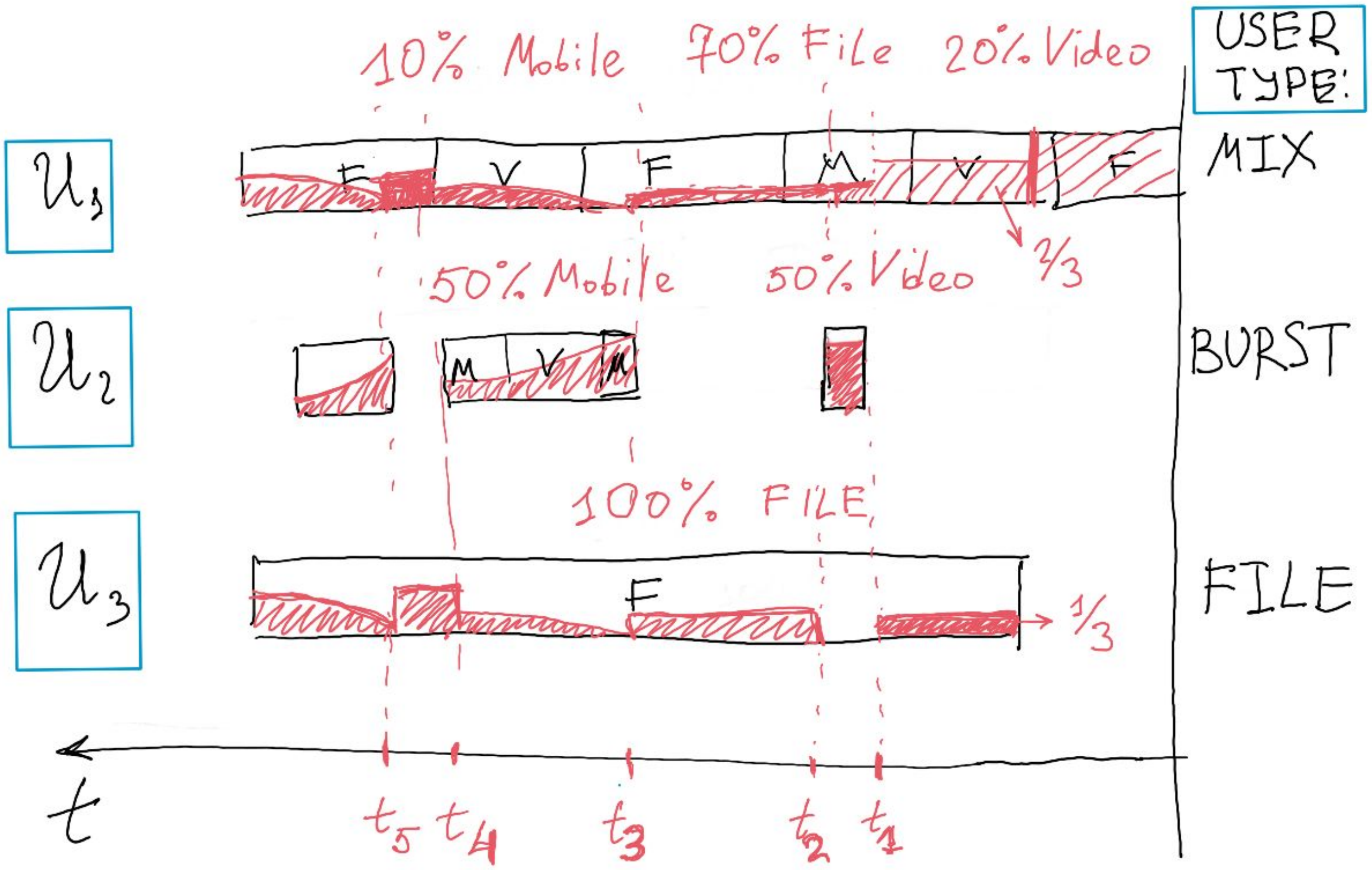
Versatile Enhancement in Dynamic Resource Optimization in 5G networks

George Ovchinnikov
Skolkovo Institute of Science and Technology

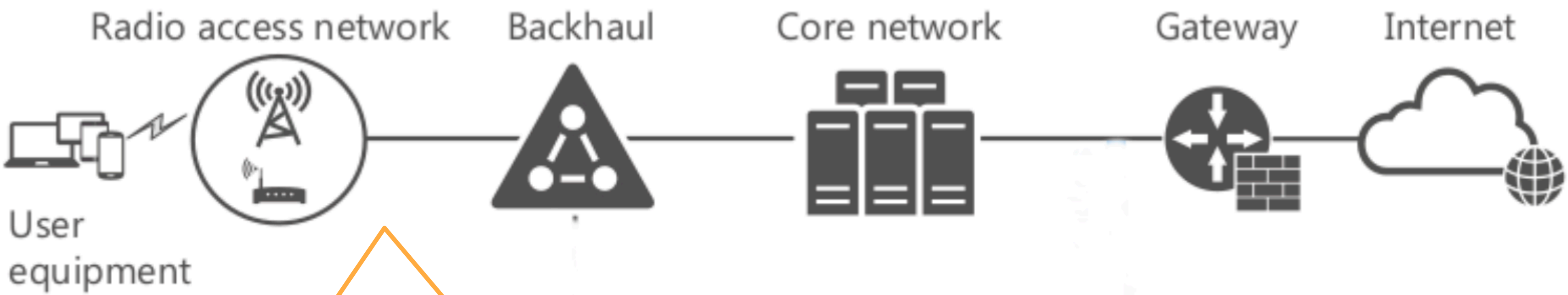
Different types of traffic have different QoS requirements

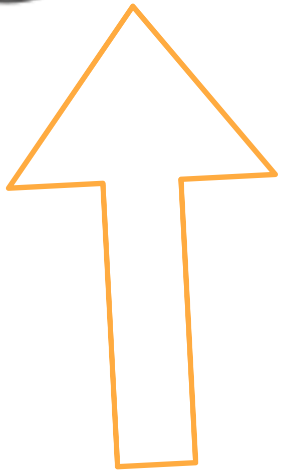


In the beginning



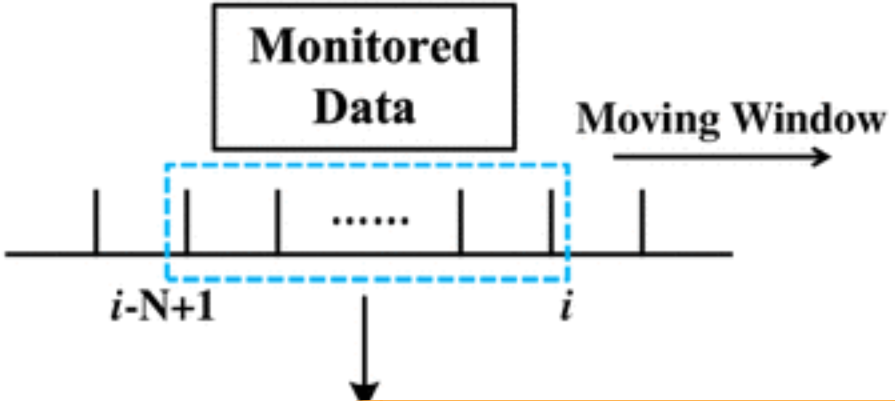
Project structure





We are here

Time-series analysis



Different QoS for different traffic types:

1. eMBB
2. mMTC
3. URLLC

Available metadata.

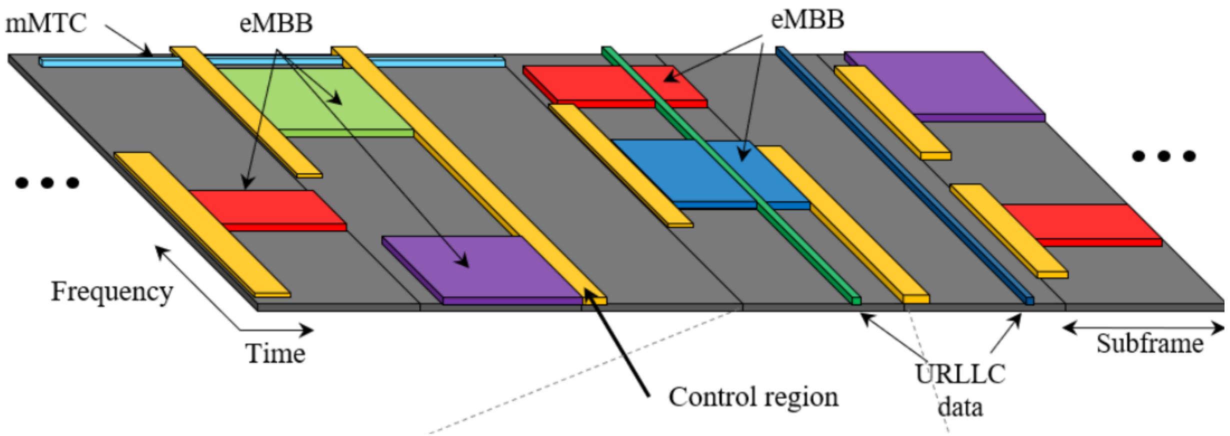
<u>May be a different models for different traffic types.</u>

1. Different prior distributions and their parameters for different users, estimated from history.

2. Statistical hypothesis testing - determining traffic type (burst of full buffer).

3. Usage of this information in scheduling rule

Packet level scheduling

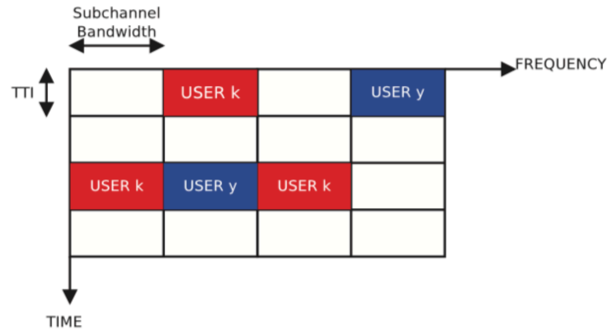


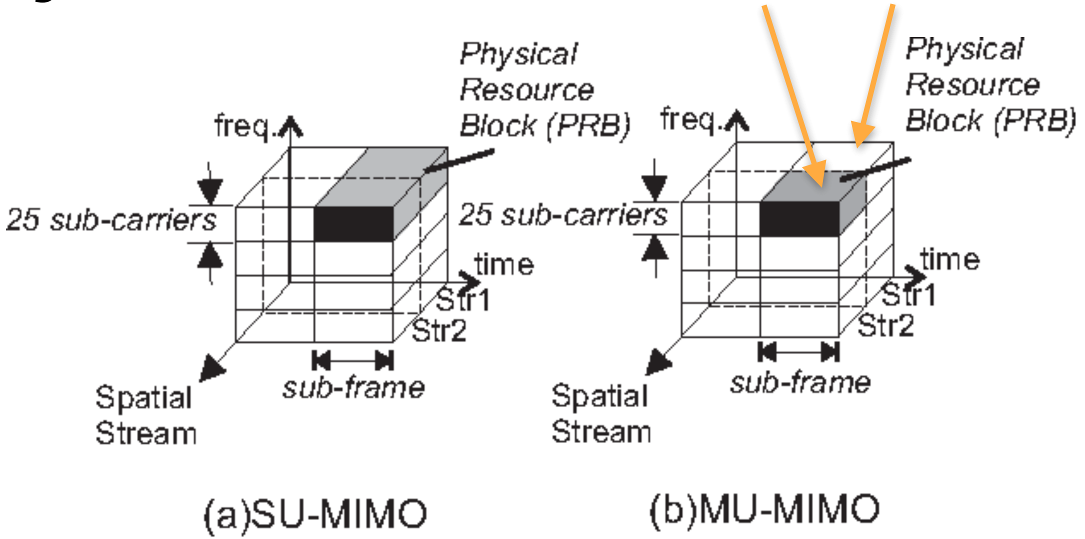
Variation of ***assignment problem***. Formally:

1) Time is discrete.
2) There is a buffer.
3) *packets* arrive at the buffer over time. Each packet has an integer release time, an integer deadline, and a positive real value (given by scheduling rule. To achieve fairness we prioritise packets for certain users).
4) In each time step, at most N packets can be sent out of the buffer.
5) The objective is to maximize the total value of the packets sent by their respective deadlines in an online manner.

Project structure

Allocated to different users





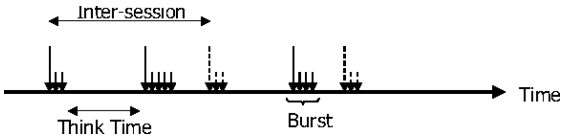
Scheduling for single-user case

Scheduling for single-user multirank case

Scheduling for multi-user case

Beyond generalised PF

$$\max P^k(t) = \frac{(r_k(t))^{\beta_i}}{(R^k(t))^{\alpha_i}}$$



Objective is to maximise perceived throughput and minimise delay.

PF gives us prioritisation of users in a way which usually balances between channel utilisation and fairness.
Not enough for burst traffic.
PF == “Instant view” of state, history averaged out.

We will look at packets history, it is distribution, moments, etc.!

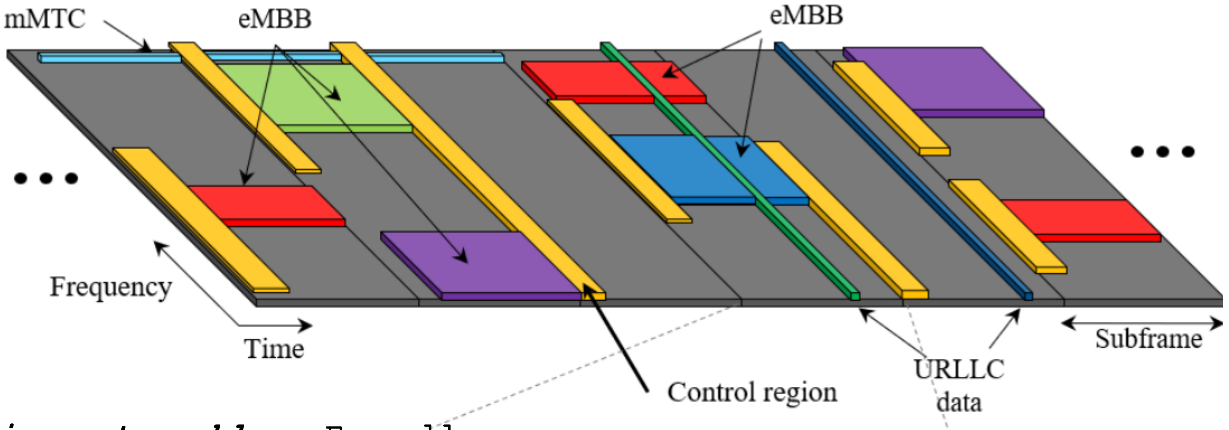
Packets in history for single user:

White == no data.
Color == data (of different types by different colours).

This is a vector. We will user vector for each user [0, 1, 0, 1, …] as a feature for machine learning to better balance users.

This is a distribution. It has an expectation, dispersion, etc.

Back to packet level scheduling



Variation of ***assignment problem***. Formally:

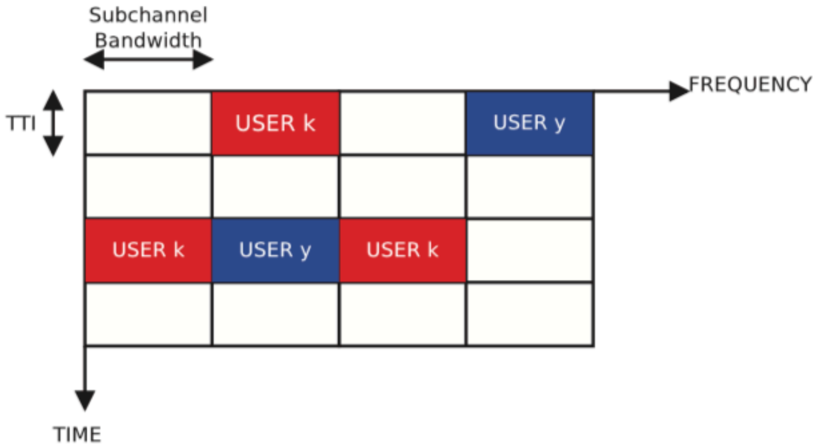
1) Time is discrete.
2) There is a buffer.
3) *packets* arrive at the buffer over time. Each packet has an integer release time, an integer deadline, and a positive real value (given by scheduling rule. To achieve fairness we prioritise packets for certain users).
4) *NEW* We have a-priory model of traffic usage patterns. We compute parameters of the model and to solve assignment problem using statical clues.
5) In each time step, at most N packets can be sent out of the buffer.
6) The objective is to maximize the total value of the packets sent by their respective deadlines in an online manner.

Different QoS for different traffic types:

1. eMBB
2. mMTC
3. URLLC

SINGLEUSER CASE

We utilise statistical information about channel and user behaviour to set up weights in assignment problem.

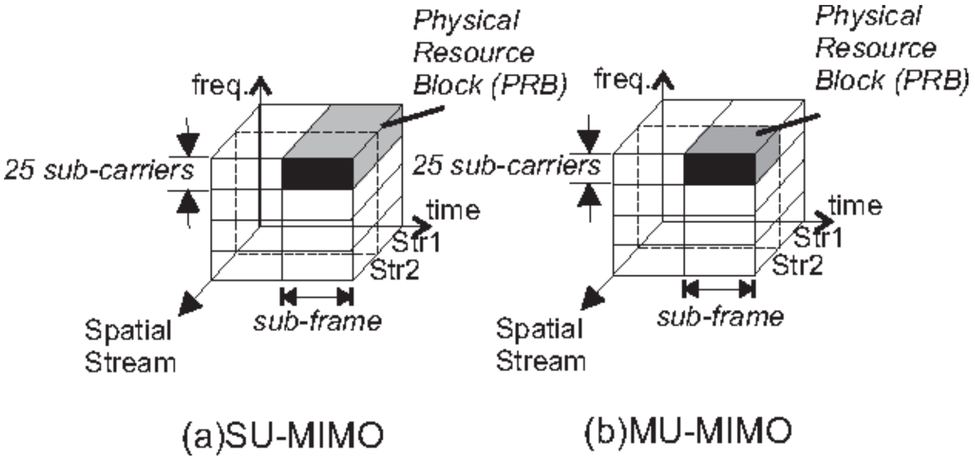


Things to try:

1. Hungarian algorithm (could be too slow)
2. Greedy algorithms (may be not optimal)
3. Clustering of users -> Hungarian algorithm
4. Hungarian algorithm with re-initialization from the previous case
5. Stochastic fixed-time methods?

SU MULTIRANK and MULTIUSER CASE

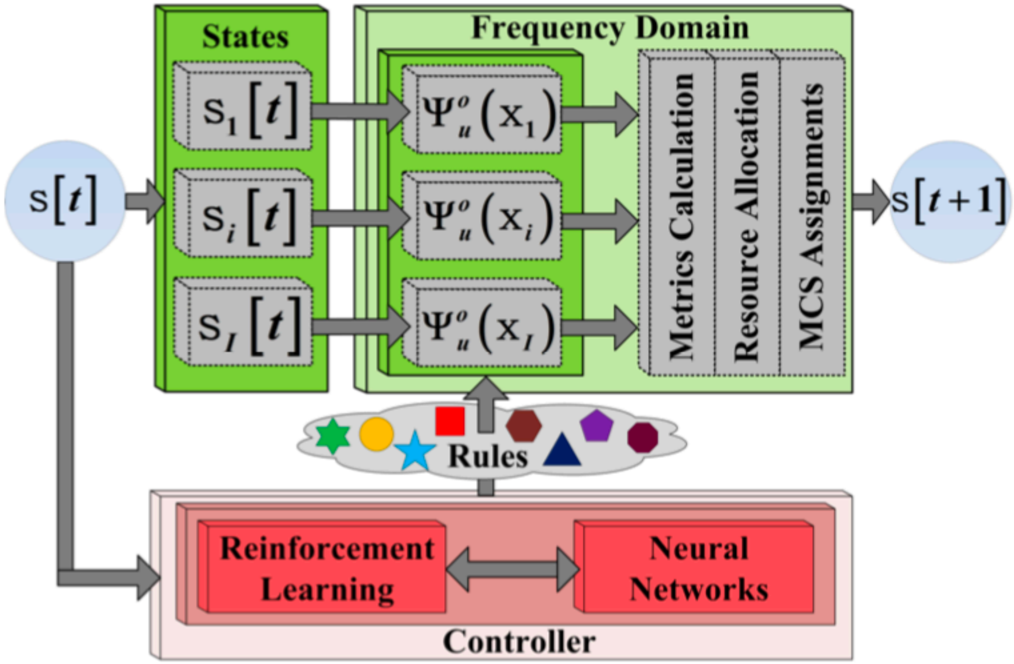
We utilise statistical information about channel and user behaviour to set up weights in assignment problem.



While mathematical formulation of scheduling is still the same form of the resource allocation problem, this time it is is more difficult, due to large search space size.

We will try to speed up heuristics from SU case.

Use neural networks



Pre-trained feature extractors ("Merlin paper")