Virtual network function forwarding graph embedding in 5g and post 5g networks

Yassine Hadjadj-Aoul

VIRTUAL NETWORK FUNCTION FORWARDING GRAPH EMBEDDING IN 5G AND POST 5G NETWORKS

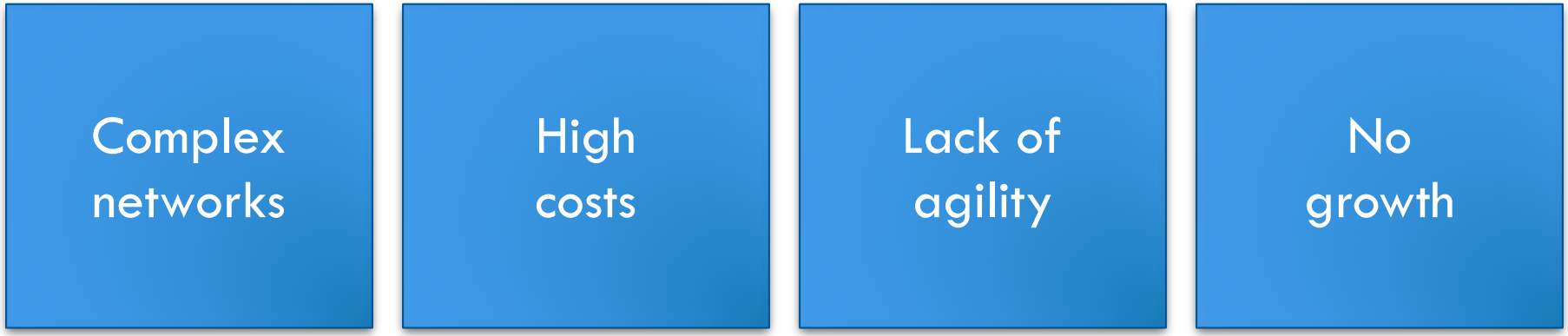
Yassine HADJADJ-AOUL
Associate professor at Univ. Rennes 1
IRISA Lab./INRIA Dionysos team-project

December 23, 2019

PLAN

- What are the challenges facing network operators today?
- Challenges related to network slicing
- On using Deep Reinfocement Learning for network slicing
- Conclusions

WHAT ARE THE CHALLENGES FACING NETWORK OPERATORS TODAY?

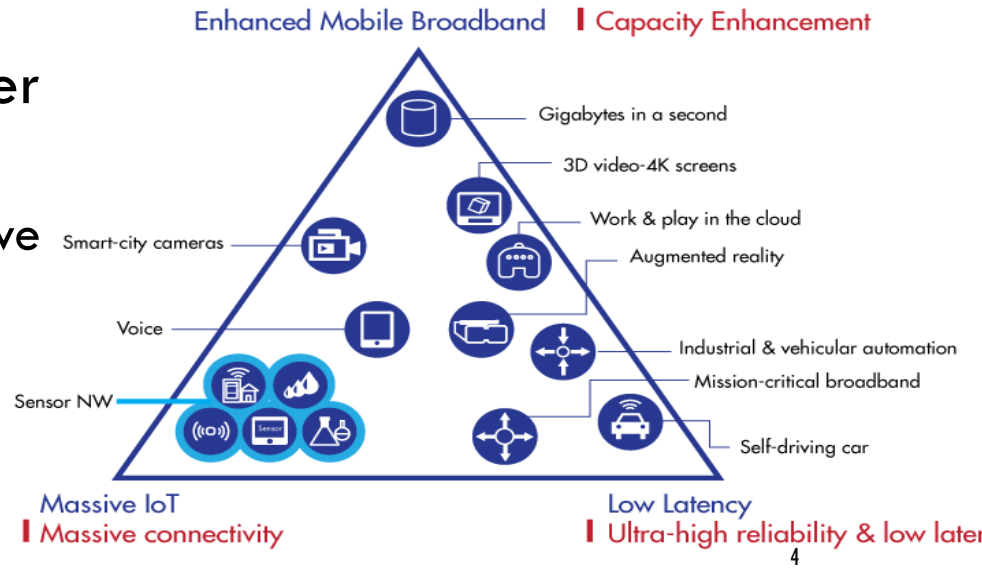


WHAT ARE THE CHALLENGES FACING NETWORK OPERATORS TODAY?

Ever-increasing infrastructure complexity

- Diversification of services (IoT, Smart cars, …)
  - Very diverse needs in terms of QoS (SLA)
  - Within the same network infrastructure
- Limits of the human being to manage a large number of equipments (10K, 100K devices, …)
  - Very high risk of mistakes, the cost of which can be prohibitive
  - Very slow service provisioning (not automated)

Complex networks



Source: ETRI graphic, from ITU-R IMT 2020 requirements

WHAT ARE THE CHALLENGES FACING NETWORK OPERATORS TODAY?

High costs

Very high investment (CAPEX) cost

- Equipment excessively expensive to purchase

Very high operational (OPEX) cost

- Significant operational costs with the human factor at the different levels of control and supervision

WHAT ARE THE CHALLENGES FACING NETWORK OPERATORS TODAY?

Lack of agility

Lack of agility

- Equipment that can hardly be adapted to the needs and of which any update is complex and not always possible
- Scaling is not always possible and oversizing is costly (unlike the Cloud)

WHAT ARE THE CHALLENGES FACING NETWORK OPERATORS TODAY?

No growth

It is very difficult to grow

- Renting infrastructure is no longer as profitable
  - Difficult to be profitable when you don't decide on the rates
  - It's difficult to get a return on investment when having a continuous evolution of the infrastructure …
- Operators are not part of the delivery chain of the service, which is very profitable (e.g. CDN)

OPERATORS' NEEDS

Automated network infrastructure

- Self configuration, self healing, self scaling, self *, …

Supporting current and future services within the same infrastructure

- With very diverse constraints (latency, bandwidth, loss, CPU, FPGA, …)

Softwarization of the network and the services

- Ability to lease infrastructure to third parties without compromising the network and its efficiency
- Higher programmability (e.g. Yang, P4 …)

Being part of the service delivery value chain

Complex networks

High costs

Lack of agility

No growth

ENABLING TECHNOLOGIES

Automated network infrastructure
- Towards autonomic networks or even Zero-touch networks
- Intent driven networking

Supporting current and future services
- On demand service composition

Being part of the service delivery value chain
- Multi-access Edge Computing (MEC)

Softwarization of the network and the services
- Software defined networks (SDN)
- Network Function Virtualization (NFV)

High level network description and programming
- E.g. Yang, P4

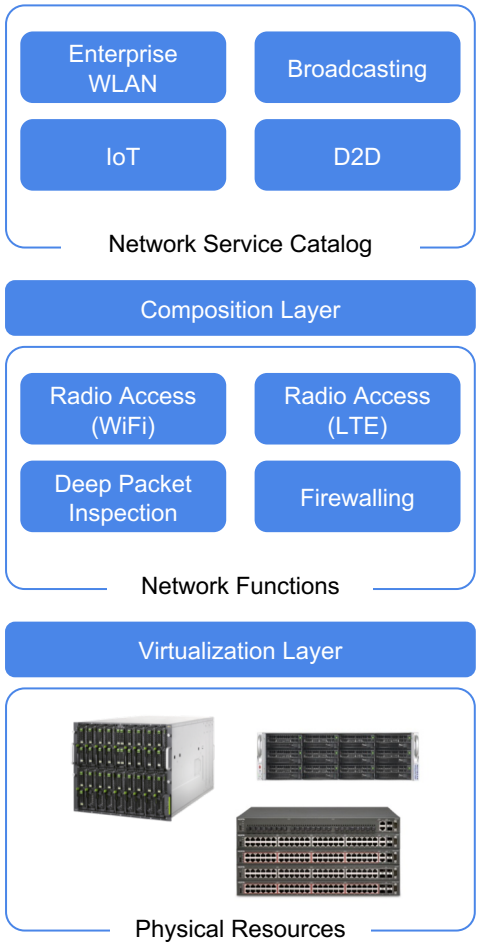
Complex networks

High costs

Lack of agility

No growth

INSPIRATION CAME FROM THE CLOUD



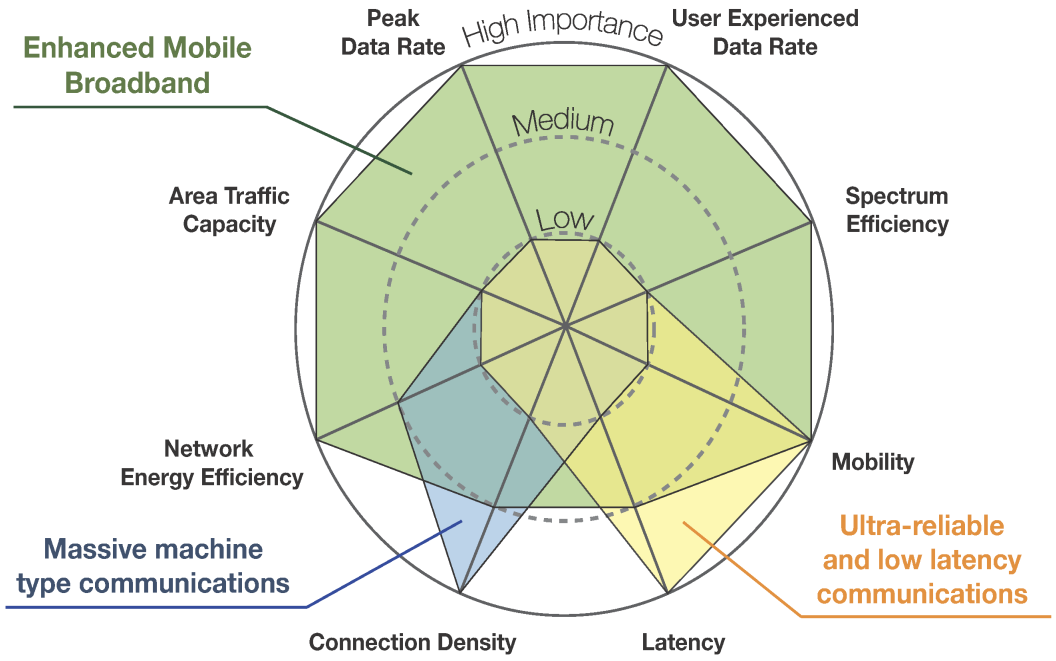
The four NFV Pillars

- Virtualized network functions
- Reuse hardware across tenants
- Chain virtual functions
- Dynamic scaling of network services

WHERE'S 5G IN ALL THIS?

5G is trying to integrate several of these building blocks to meet the needs of the NOs

- For this purpose the concept of network slicing is introduced.



MY DEFINITION OF NETWORK SLICING

Network slicing is the process that allows a dynamic and personalized sharing of network operators' infrastructure

Target: Carrier grade virtual network

WHY ARE OPERATORS INTERESTED IN NETWORK SLICING? (1)

Network slicing is seen as an opportunity for NO to get a lot more from their infrastructure

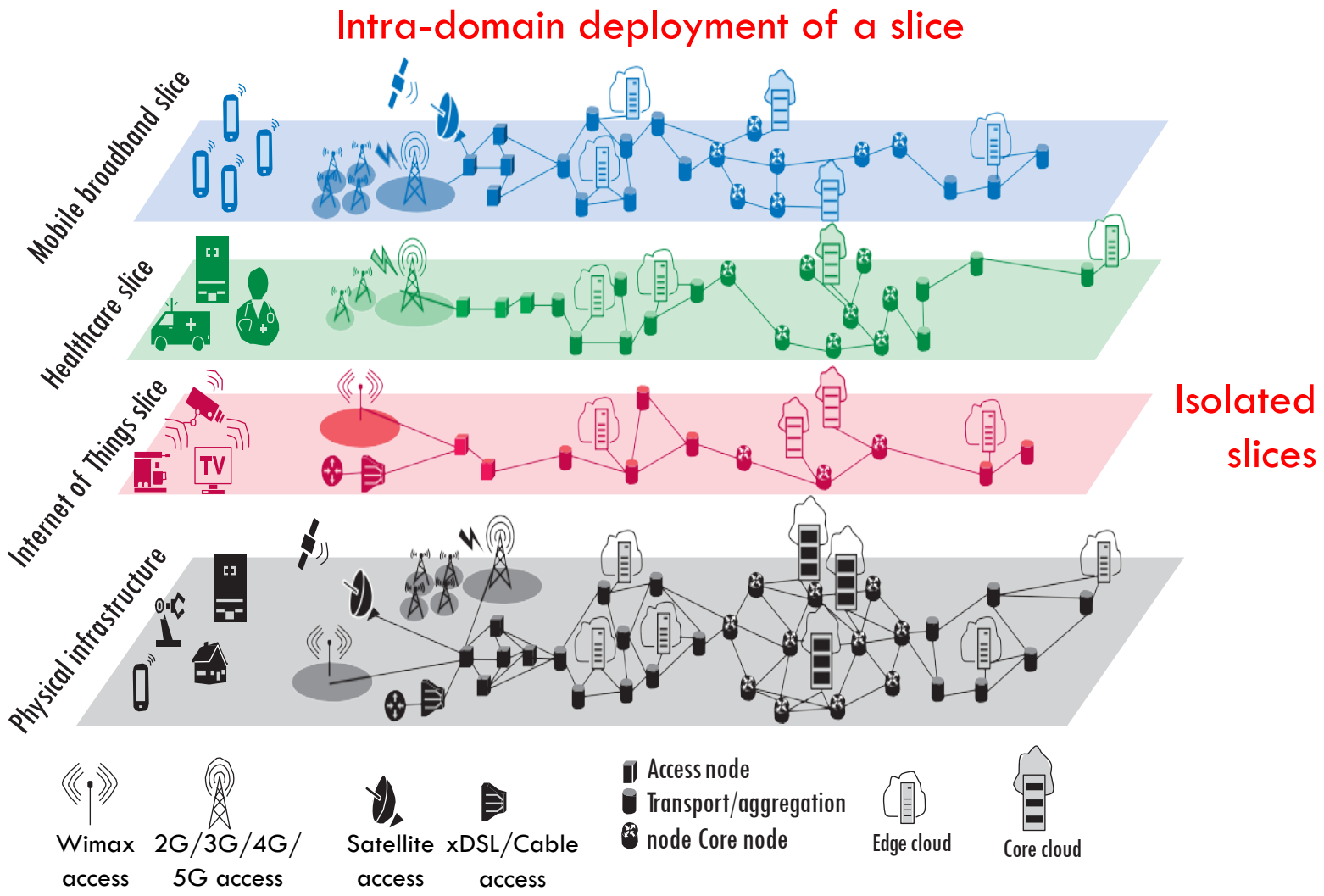
- by offering numerous dedicated virtual networks specifically tailored towards a service or customer
- by operating on virtual network architecture and borrows from the principles behind SDN and NFV

WHY ARE OPERATORS INTERESTED IN NETWORK SLICING? (2)

Network slicing allows to simultaneously accommodate a wide range of services

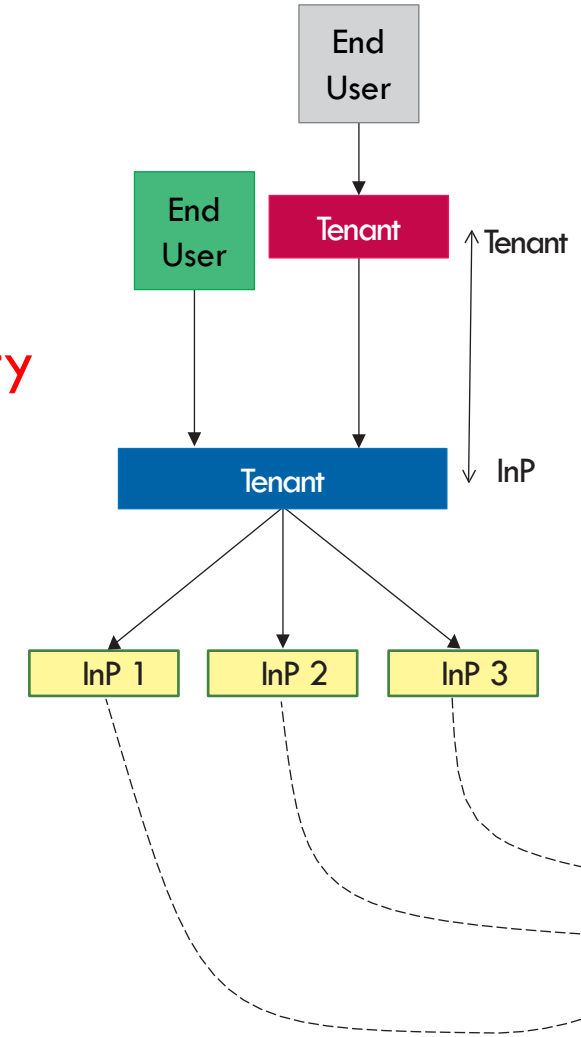
- over a common network infrastructure

May support new services on-demand and in near real-time

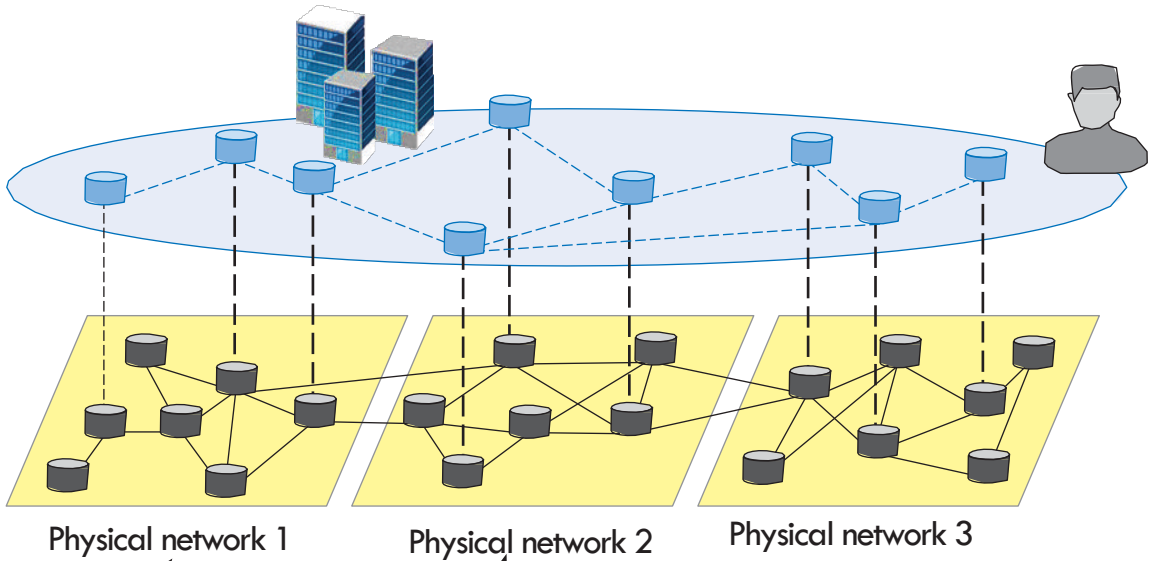


NETWORK SLICING: AN ARCHITECTURE

Recursivity



Multi-domain deployment of a slice



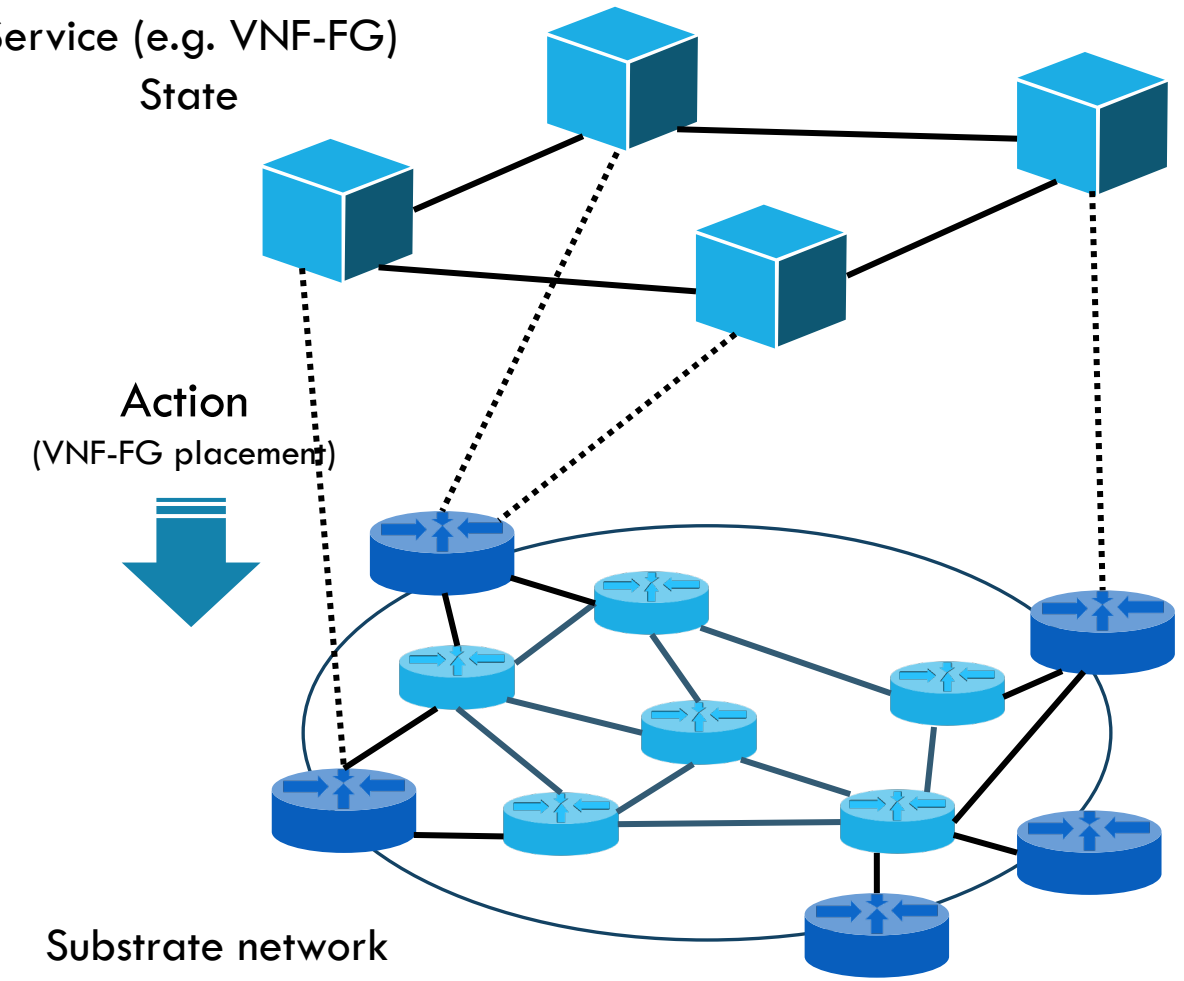
IN PRACTICE, WHAT DOES SLICING A NETWORK CONSIST OF? (1)

1 Most simple form

- Placement of services consisting in one VNF
  - Offline vs online problem (bin packing problem)
    - May consider 1 or several metrics for the placement (e.g. latency, load, reliability, …)
  - NP-complete problem

2 More advanced form (more complex)

- Placement of services in a VNF-FG form
  - Involves not only the placement of VNFs but also addressing a routing problem
    - Addressing the VNFs placement then the routing … or simultanuously
  - Need to consider several metrics (QoS requirements)
  - Intra-domain or multi-domain placement



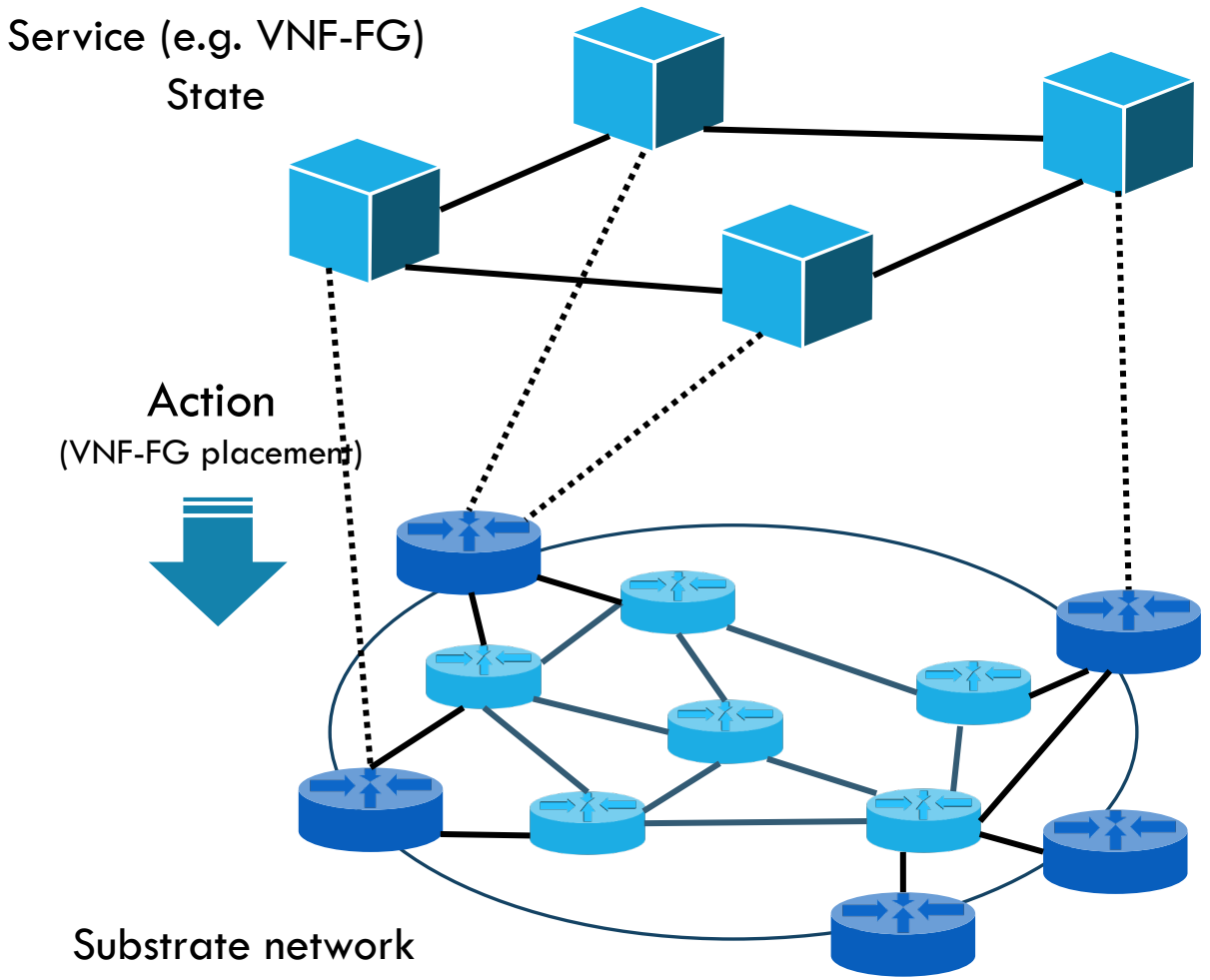
IN PRACTICE, WHAT DOES SLICING A NETWORK CONSIST OF? (2)

3 More advanced form

- Placement and scalability of services
  - Involves the placement of services while automatically managing their scalability
    - Determining the number of instances of each VNF?
    - Could be also managed at runtime, which make the placement even harder

4 Most advanced forms

- Slices could be isolated (waste of resources) or may share the resources (risk of outage)
- Consists in considering the sharing of VNFs between different services
- Considering microservices, …



WHAT METRICS ARE CONSIDERED FOR PLACEMENT?

This naturally depends on the problem being addressed (SLA) …

- Reliability
  - Loss
  - K-connectivity, average connectivity, …
- Service requirements
  - Bandwidth
  - System requirements (CPU, RAM, Disk, FPGA, …)
  - QoS/QoE
  - Load balancing
- Scalability
- Energy/power saving

Some problems require addressing one metric and others several metrics at a same time … with the risk of a combinatorial explosion.

SERVICES PLACEMENT: A WELL STUDIED PROBLEM!

The problem of placement is an old and well-studied problem

- Many papers in the Cloud context
- Conventional service placement comes down to a problem of bin-packing or knap-sack
  - NP-hard problem
- Realistic placement of services, like the VNF-FG placement is much more complex
  - Services are composite, since they include several sub-services, and multi-constrained
    - Must be added the multiple constraints on the links

Less attention has been paid to the placement of VNF-FG, which is actually a fairly recent issue

CLASSIFICATION OF EXISTING APPROACHES FOR THE PLACEMENT

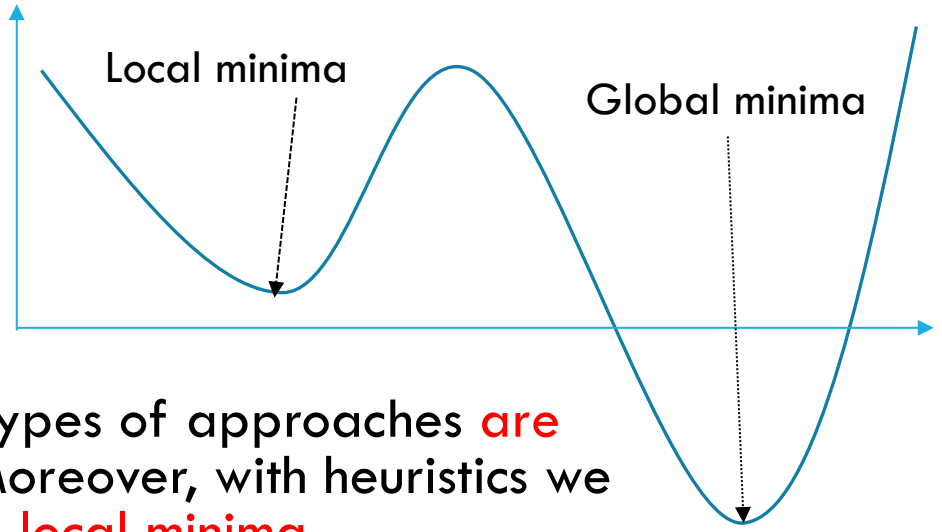
1. Mathematical optimisation-based approaches
   - Most of the paper fall in this category
   - Use of: Integer Linear Program (ILP) or Mixed ILP (MILP) …
     - Integer Programming is an NP-complete problem. So:
       - There is no known polynomial-time algorithm
       - Even small problems may be hard to solve
         - Propose more efficient heuristics for solving the problem, so they fall into another category.

Main limitation:

Some parameters are only obtained during run-time (latency and loss) which makes these approaches sometimes ineffective in a real context.

CLASSIFICATION OF EXISTING APPROACHES FOR THE PLACEMENT

2. Heuristics-based approaches
   - Most of the paper fall in this category
   - Use generally : a two step approach
     - Placing VNFs using traditional algorithm (First Fit, Best Fit, nearest search procedure, …)
     - Then placing VLs (using Shortest path "SP", K-SP, …)
     - Very fast, effective and deal with very large problems
       - For some industrials, this is the best solution



Main limitation:

In systems where constraints and objectives are changing, these types of approaches are not very suitable since they generally require a total redesign. Moreover, with heuristics we have a rapid convergence at the price of the risk of sticking at a local minima.

CLASSIFICATION OF EXISTING APPROACHES FOR THE PLACEMENT

3. Metaheuristics-based approaches

- Use generally : a two step approach
  - Placing VNFs using an evolutionary, greedy, … algorithm
  - Then placing VLs using Shortest path ”SP”, …) or using a metaheuristic
  - Slow, very effective and deal with very large problems
    - Explore all solutions, or only feasible solutions (faster with risk of stacking at a local optimum)
    - Can be used in a real or simulated system
  - With enough time this may converges to global optimum
- But …
  - As the fitness (cost) function is function of VNFs + VLs placement … it comes to placing both at the same time …

CLASSIFICATION OF EXISTING APPROACHES FOR THE PLACEMENT

3. Metaheuristics-based approaches

Features and main limitation:

Metaheuristics make it possible to respond effectively to the problem VNF-FG placement. They can very easily integrate new objectives or constraints without reconsidering the solution, unlike heuristics. It should also be noted that there is some research work that shows the convergence of these algorithms towards the optimum under certain conditions.

However, to address a new placement you almost always need to start from the beginning … as there is no learning

CLASSIFICATION OF EXISTING APPROACHES FOR THE PLACEMENT

4. Learning-based approaches
   - Only very few approaches
   - Use generally : a two step approach but … the reward function concerns VNFs + VLs placement, which means that we are addressing both at the same time.
     - Very slow
       - For some industrials, this is the best solution

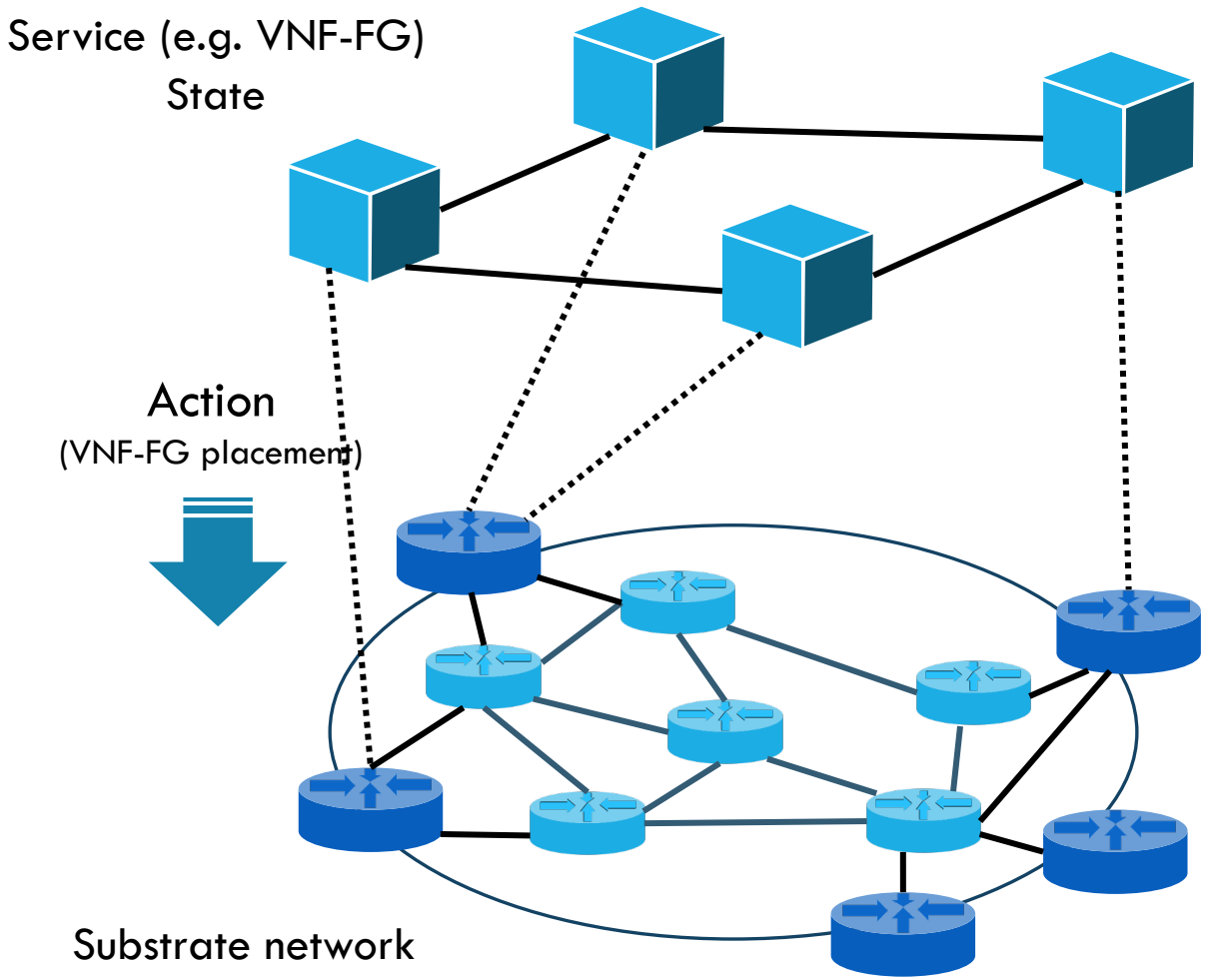
Main limitation:

Efficient approaches still need to be developed.

OUR OBJECTIVE

Problem 2-~3-~4

- Placement of services in the form of VNF-FG
- Shared resources
- Intra and multi-domain
- Constrained services' placement
  - Networking requirements: bandwidth, latency, loss, …
  - System requirements: CPU, RAM, DISK, FPGA, …



MANY COLLABORATIONS WITH INDUSTRY …

Orange

- Yue Li (MEC)
  - Heuristics – Water filling, Control theory
- Jean-Michel Sanner (Controllers' placement)
  - Heuristics, Evolutionary Algo.
- Farah Slim (Online VNFs placement)
  - Heuristics, Evolutionary Algo.

TDF

- Imad Alawe (Times series forecasting)
  - Deep learning

INRIA

- Hamza Ben Ammar (Cache placement)
  - GRASP

Nokia Bell Labs

- Anouar Rkhami (VNF-FG placement) … ongoing
- Quang Pham Tran Anh (VNF-FG placement + routing) – Post doc
  - Evolutionary algo., Deep learning

EXFO

- Reliable placement (will start soon)

CHALLENGE IN VNF-FG PLACEMENT

Extremely large number of possibilities of placement (very large action space)

- Difficulty in finding an optimal placement (i.e. NP-hard problem), excepting for very small instances

Limits of existing work

- Optimization problems (i.e. MILP) have the limitation of not always being applicable in a real context, given the latency of resolution or unsuitability in a real context
- Problem of convergence of classical Meta-heuristics towards more or less good solutions … and no learning …
  - Previous work on Evolutionary Algorithms

Objective (short and mid-term)

- Exploring the potential of deep learning in the placement of VNF-FG

BEST ACTION SELECTION USING ML . . .

Classification of machine learning algorithms

Our focus here

Supervised learning

Knowing input X and output y (labels), we try to find f, y = f(X) (**mapping**)

Unsupervised learning

Knowing X, we **classify** them regarding some cost function

Reinforcement learning

We learn how to take actions (**policy**) to maximize a reward function

- **Value-based**: **Learn the value of being in a given state,** and taking a specific action there
- **Policy-based**: **Learn functions,** which directly map an observation to an action

« When solving a problem of interest, do not solve a more general problem as an intermediate step »

Vladimir Vapnik

DEEP LEARNING APPROACH

Machine Learning or Optimization-based?

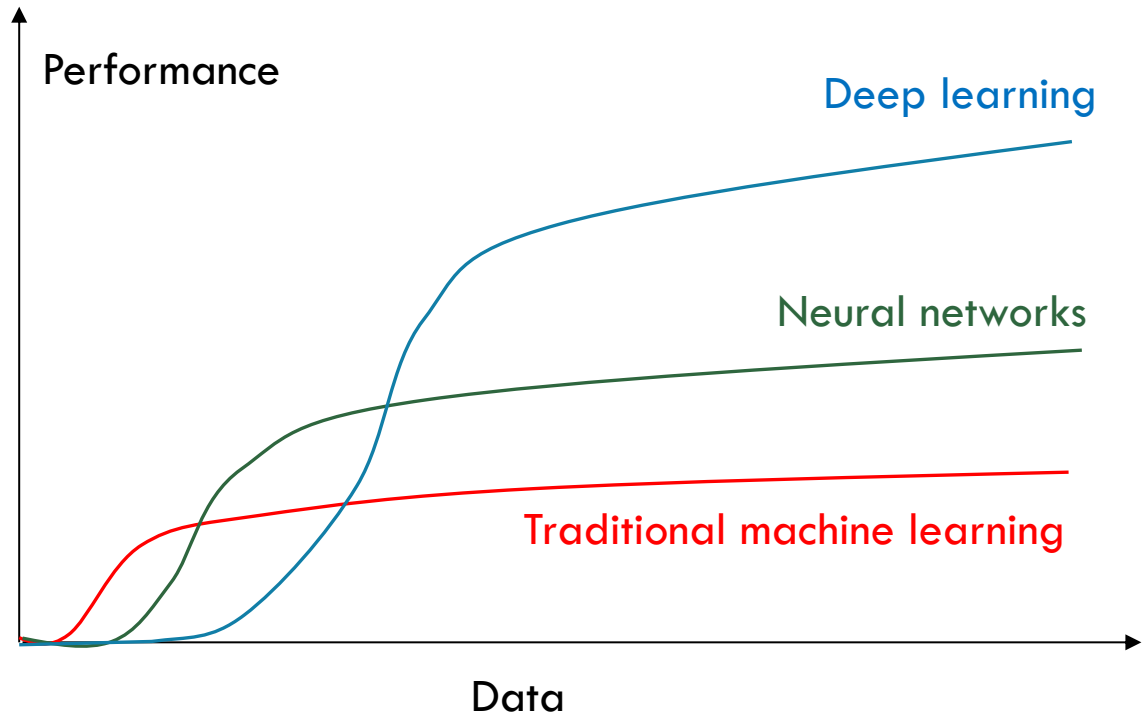
- Optimization-based approaches need accurate models
- Difficulties in determining accurate models for complex networks (multi-hop)(*)
- Machine learning addresses this by learning hidden characteristics of any network

(*) Z. Xu *et al.*, "Experience-driven Networking: A Deep Reinforcement Learning based Approach," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Honolulu, HI, 2018, pp. 1871-1879.

DEEP LEARNING APPROACH

Why go deep?

- Data dependency



WHY REINFORCEMENT LEARNING

Can be used even in the absence of labels.

Recent advances in deep reinforcement learning have shown their effectiveness to deal with the credit assignment problem

- Example: chess game,...

Recent techniques are stable

- Mix between traditional RL and supervised learning (experience reply)

Very suitable for operating in an unknown or uncontrolled environment

- Can learn and evolve in changing environments or partially observable environments (multi-domain strategy)

REINFORCEMENT LEARNING

Reinforcement learning is about mapping situations to actions to maximize a reward

- Learner is not told which actions to take, but instead must discover which actions yield the most reward

Finding the best actions involves a fairly broad exploration

- Agent must try a variety of actions and progressively favor those that appear to be best (trade-off between exploration and exploitation)
- Knowledge about the model can reduce this space
  - e.g. if we do not succeed in placing a VM of a particular size on a node, it means that we will not be able to place a machine of a larger size … not easy to learn such rules

SOME EXISTING STRATEGIES …

Q-learning algorithm is able to achieve an optimal policy

- when the state space and action space are limited (tables don't scale)

Deep Q-learning algorithm overcomes the limits of Q-learning by approximating the Q-value using a Neural Network

- But presents instabilities
  - Could be improved by improving the exploration + experience replay
- Presents some limits for problems with large action space

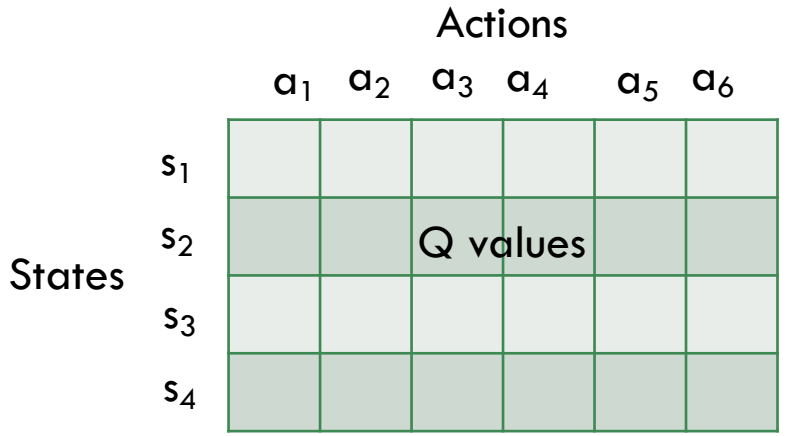
Deep Deterministic Policy Gradients (DDPG) – Google 2016 – obsoleted DQN

- More robust, and able to achieve much better scores on the standard battery of Deep RL tasks

SOME EXISTING STRATEGIES …

Q-learning

Q-Learning is a table (Q-table) of values for every state (row) and action (column) possible in the environment.



Recursive process to converge to the best Q-table

- using for example Temporal Difference Learning (TD) :

$$Q(s,a) = Q(s,a) + \alpha\big(R + \gamma \max\big(Q(s',:)\big) - Q(s,a)\big)$$

Current state

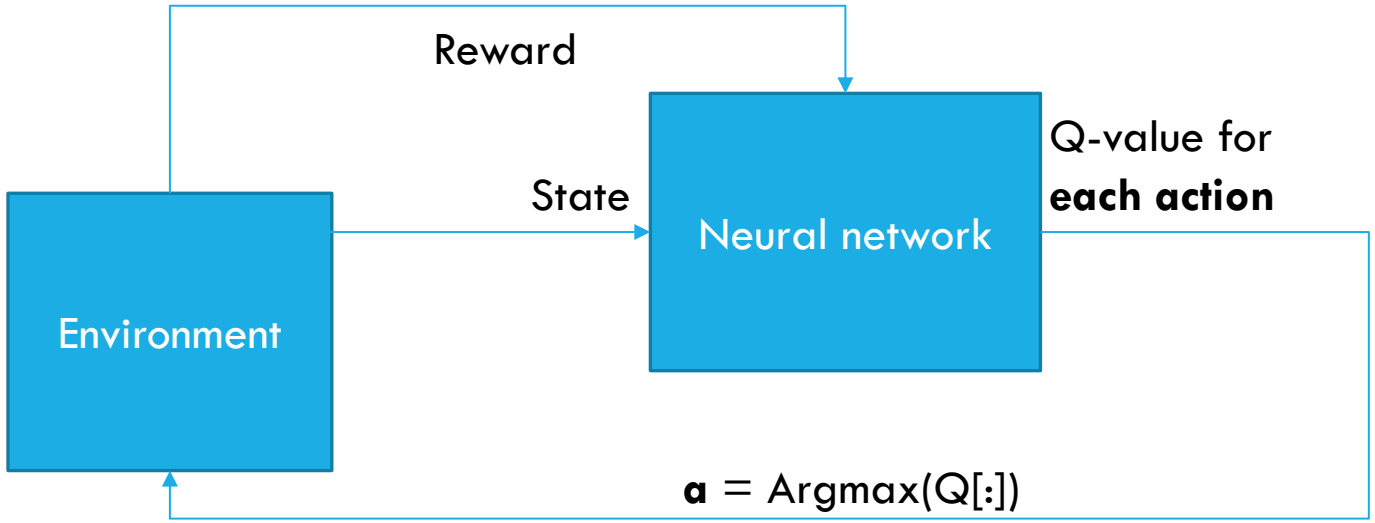
Obtained

Q-learning algorithm is able to achieve an optimal policy
when the state space and action space are limited (tables don't scale)

SOME EXISTING STRATEGIES …

Deep Q-learning algorithm overcomes the limits of Q-learning by approximating the Q-value using a Neural Network

- But presents instabilities
  - Could be improved by improving the exploration (epsilon-greedy), experience replay, target network



$$Loss = \sum (Q_{target} - Q)^2$$

- Q-target: Q-value of selected action
- Q: output of the NN

SOME EXISTING STRATEGIES . . .

Deep Deterministic Policy Gradients (DDPG) – Google 2015 – obsoleted DQN

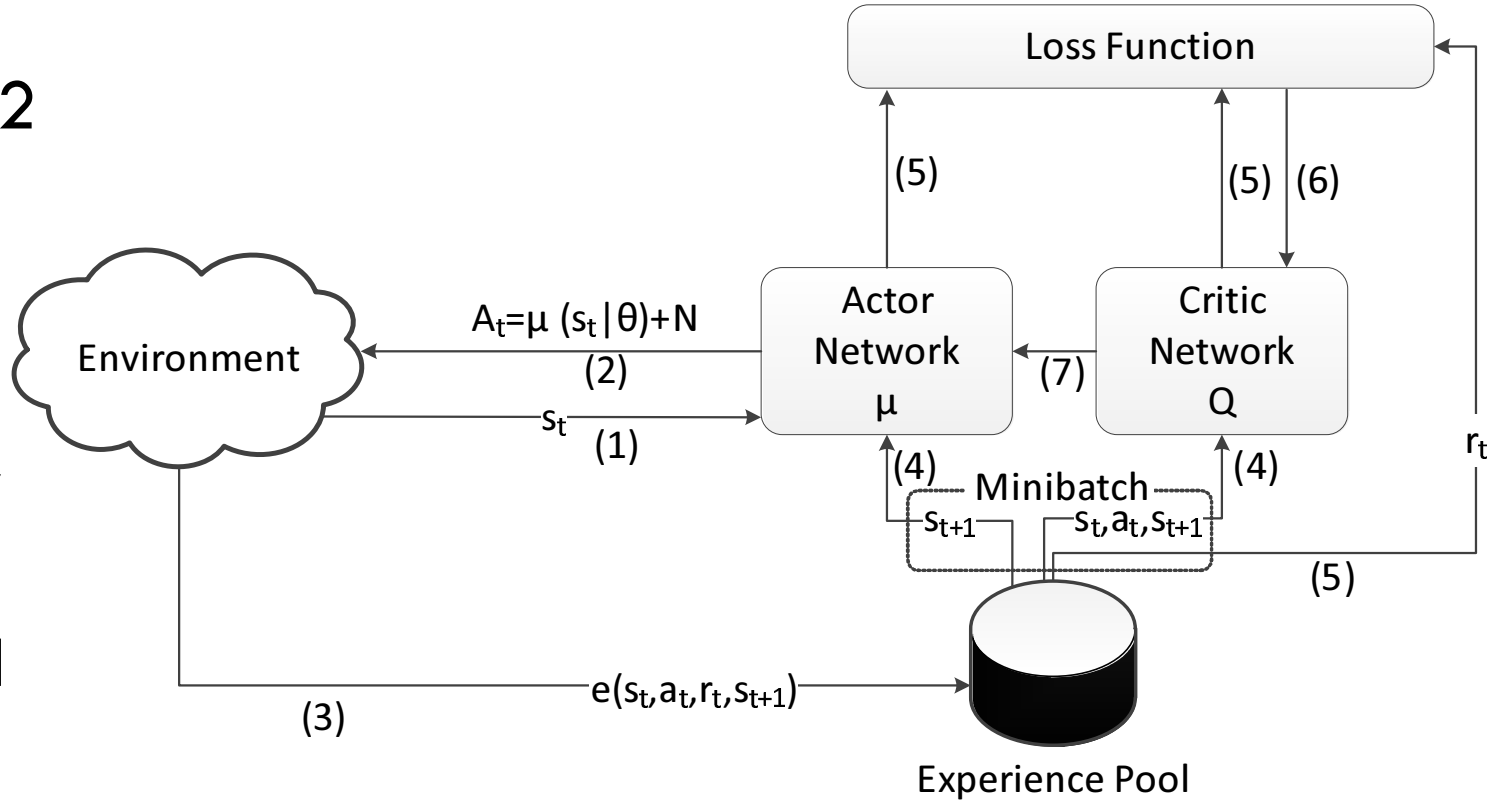
- More robust, and able to achieve much better scores on the standard battery of Deep RL tasks

FOCUS ON DDPG

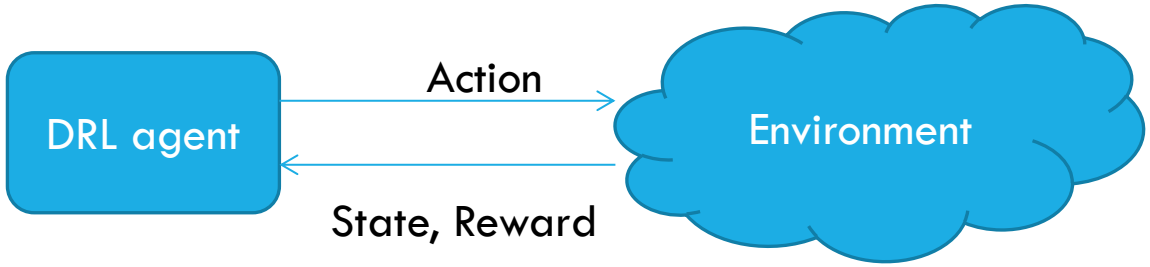
2 neural nets (Actor, Critic) + 2 target nets :

- Critic network learns Value function
  - Based on state + action
- Actor network learns the Policy

The output of the critic drives learning in both the actor and the critic



APPLYING DDPG FOR NETWORK SLICING



State: VNF-FG description [U,V]

$U=[u_{11},u_{12},\ldots,u_{1K},\ldots,u_{N'1},u_{N'2},\ldots,u_{N'K}]$: description of VNFs, u_{ij}: requirement of j^{th} resource of VNF i

$V=[v_{11},v_{12},\ldots,v_{1H},\ldots,v_{L'1},\ldots,v_{L'H}]$: description of VLs, v_{ij}: requirement of j^{th} QoS metric of VL i

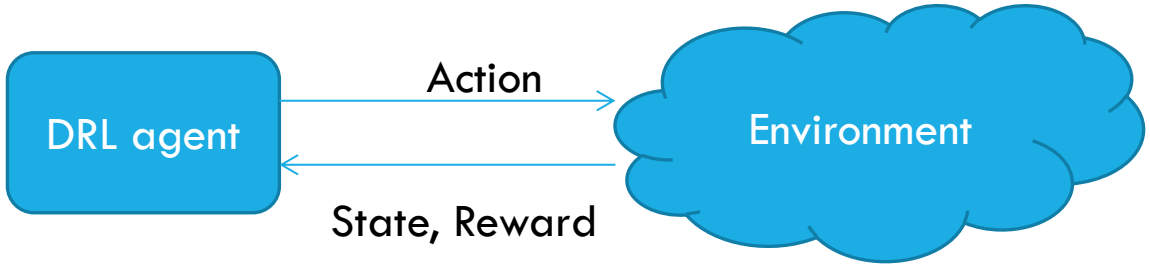
Action: $A = [a_{11},\ldots,a_{1N},\ldots,a_{N'1},\ldots,a_{N'N}, w_{11},\ldots,w_{1L},\ldots,w_{L'1},\ldots,w_{L'L}]$

a_{ij}: metric to select substrate node j for VNF i

w_{ij}: weight of link j to determine path for VL i → Each VL has unique routing policy

Reward: Acceptance Rate = (# Successful Deployed VNF-FG) / #VNF-FG

APPLYING DDPG FOR NETWORK SLICING



State: VNF-FG desc
U=[u_{11},u_{12},…,u_{1K},…
V=[v_{11},v_{12},…,v_{1H},…

Action: A = [a_{11},…
a_{ij}: metric to select
w_{ij}: weight of link j

Reward: Acceptance Rate = (# Successful Deployed VNF-FG) / #VNF-FG

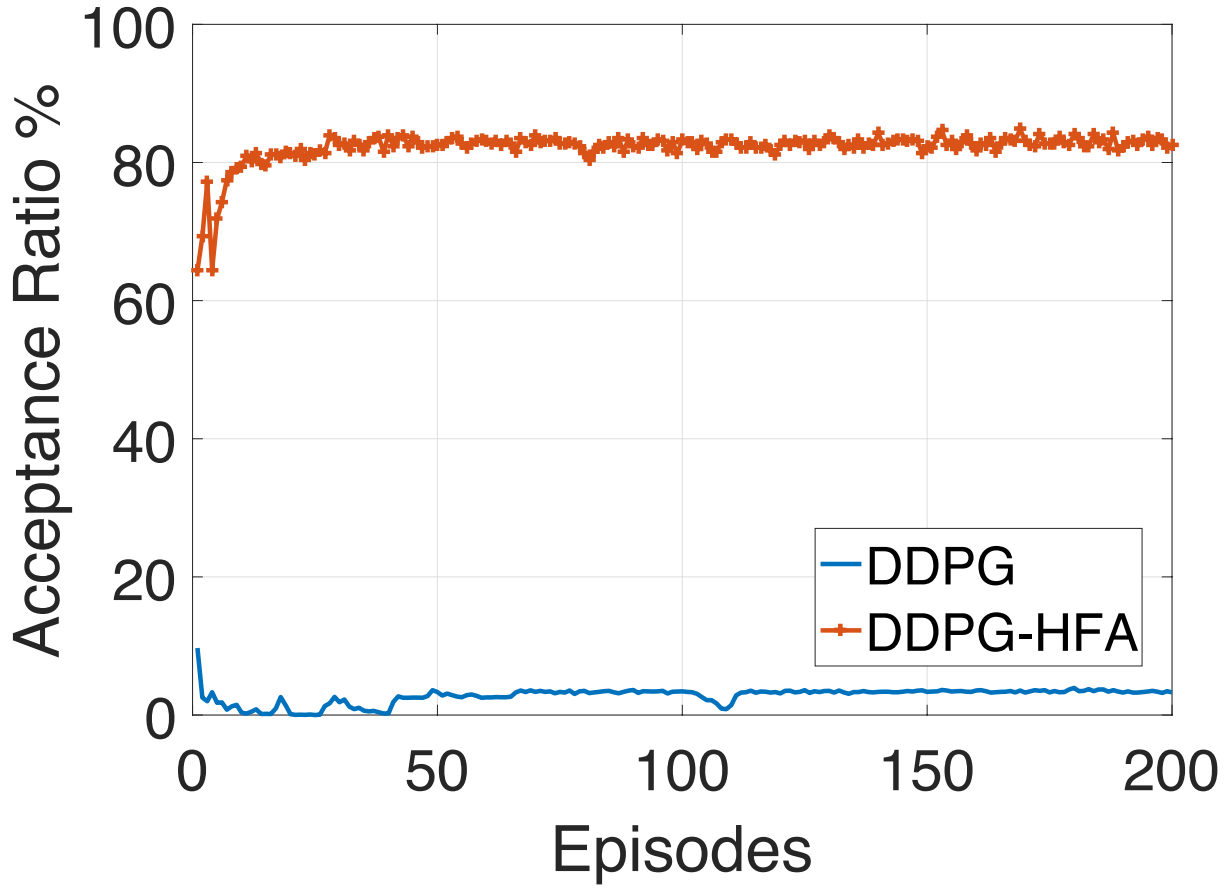
Characteristics

- Large action space
- For VNF mapping: from a_{ij} select one substrate node to host the VNF
- For VL mapping: from w_{ij} select a set of links connecting 2 VNFs

RESULTS ON THE APPLICATION OF DDPG

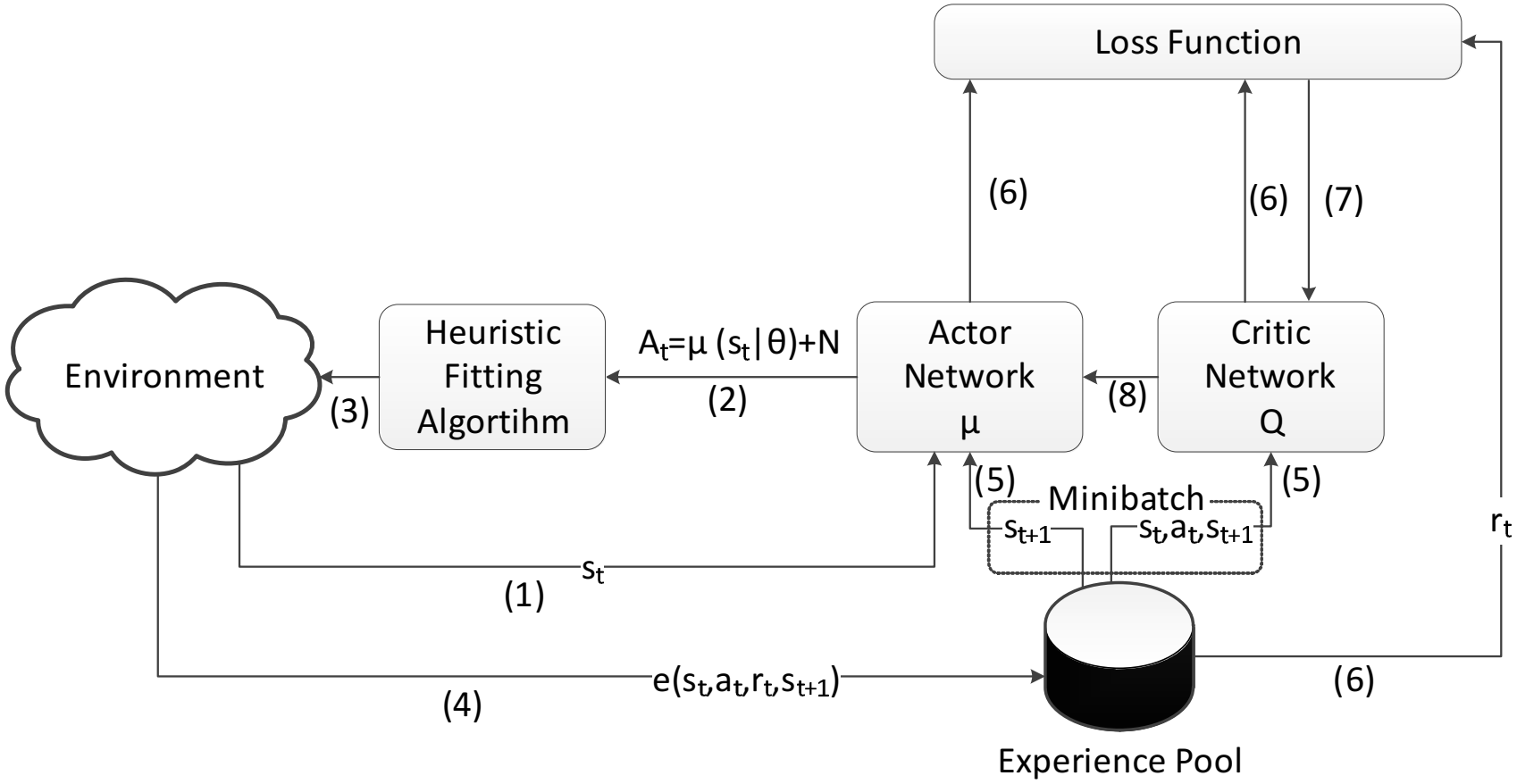
- The vanilla DDPG is not suitable for very large-scale discrete action space
- In VNF-FG embedding problem, there are constraints of resources such that some discrete actions are not feasible.

Proposing to add to DDPG a Heuristic Fitting Algorithm (HFA)



DDPG-HFA

HFA allows improving the Reward



IMPROVING EXPLORATION WITH EEDDPG

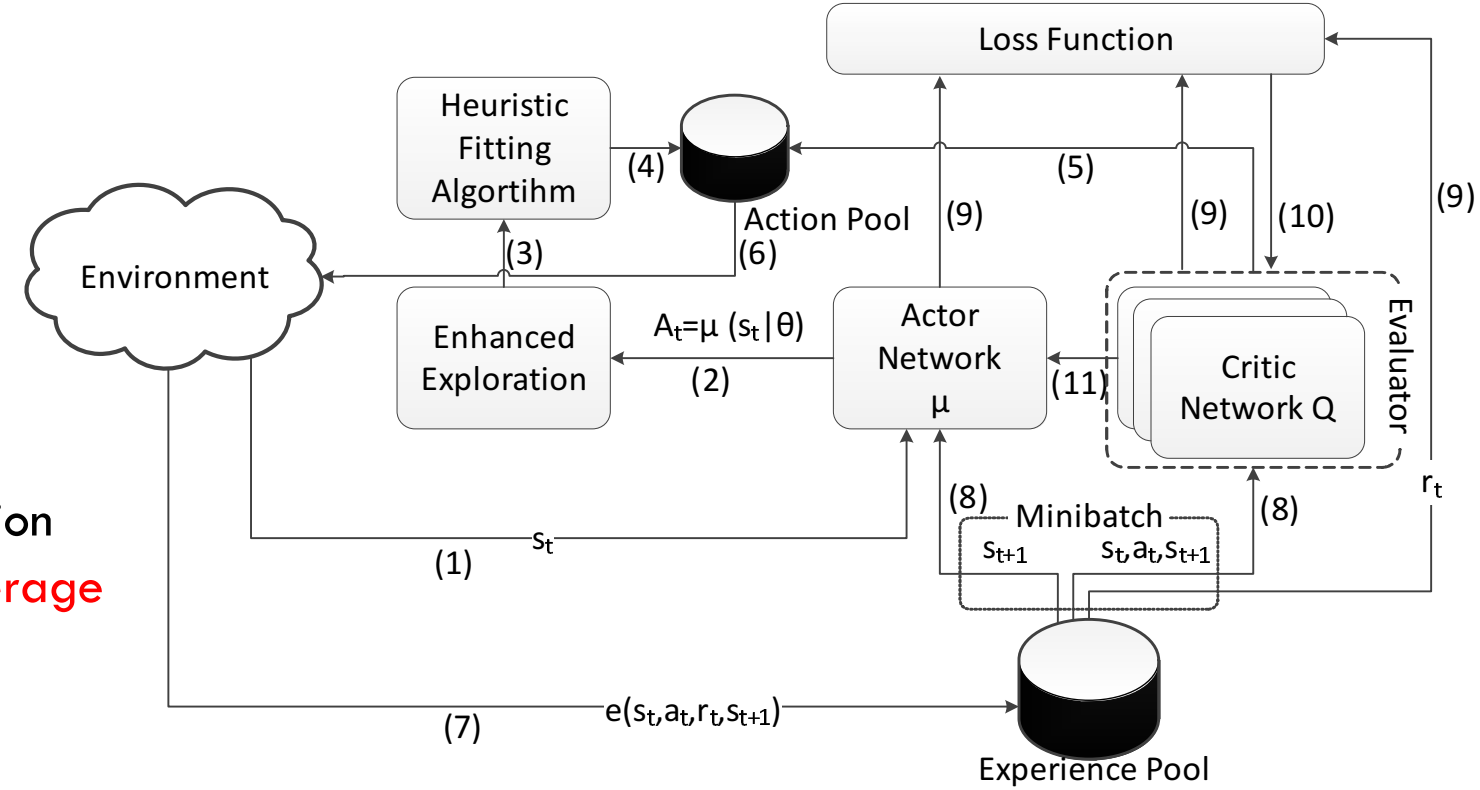
- **Noise** is added to the proto-action to create H noisy actions.
  - Each proto action is fed into HFA in order to determine the feasible allocation of VNFs.

Evaluation:

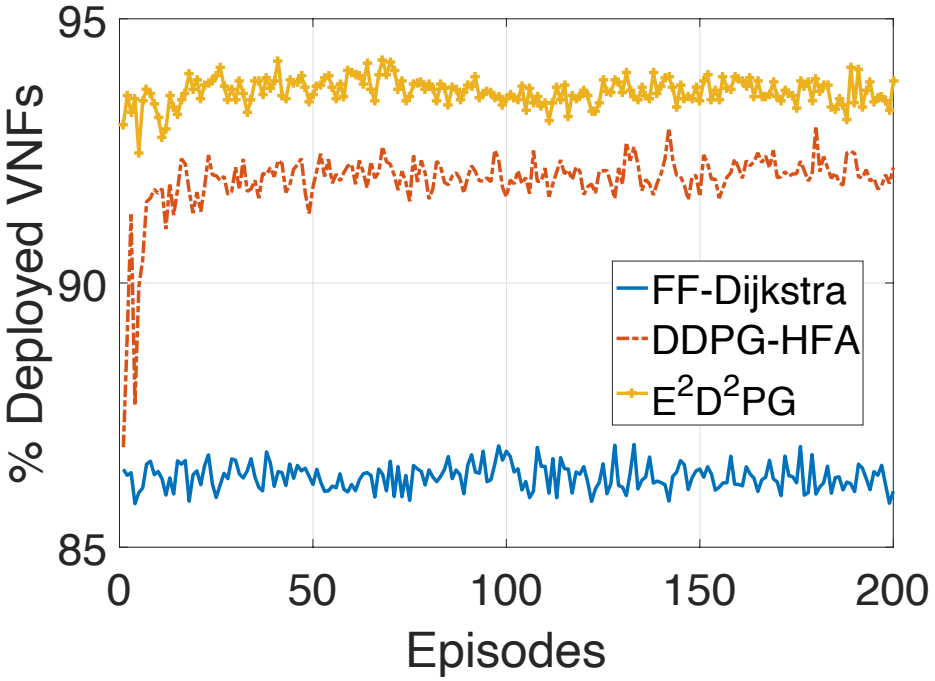
- **Multi-critic**
  - Different nets due to the arbitrary initialization
  - The best action identified (best Q value/average Q value) by the evaluator will be executed

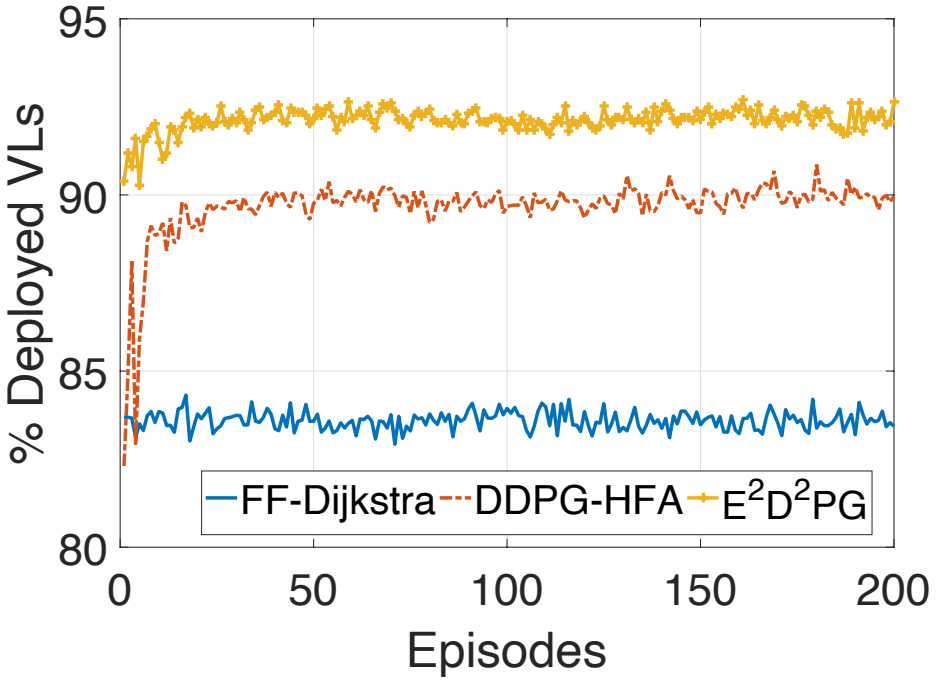
Updates of actor:

- Best critic network

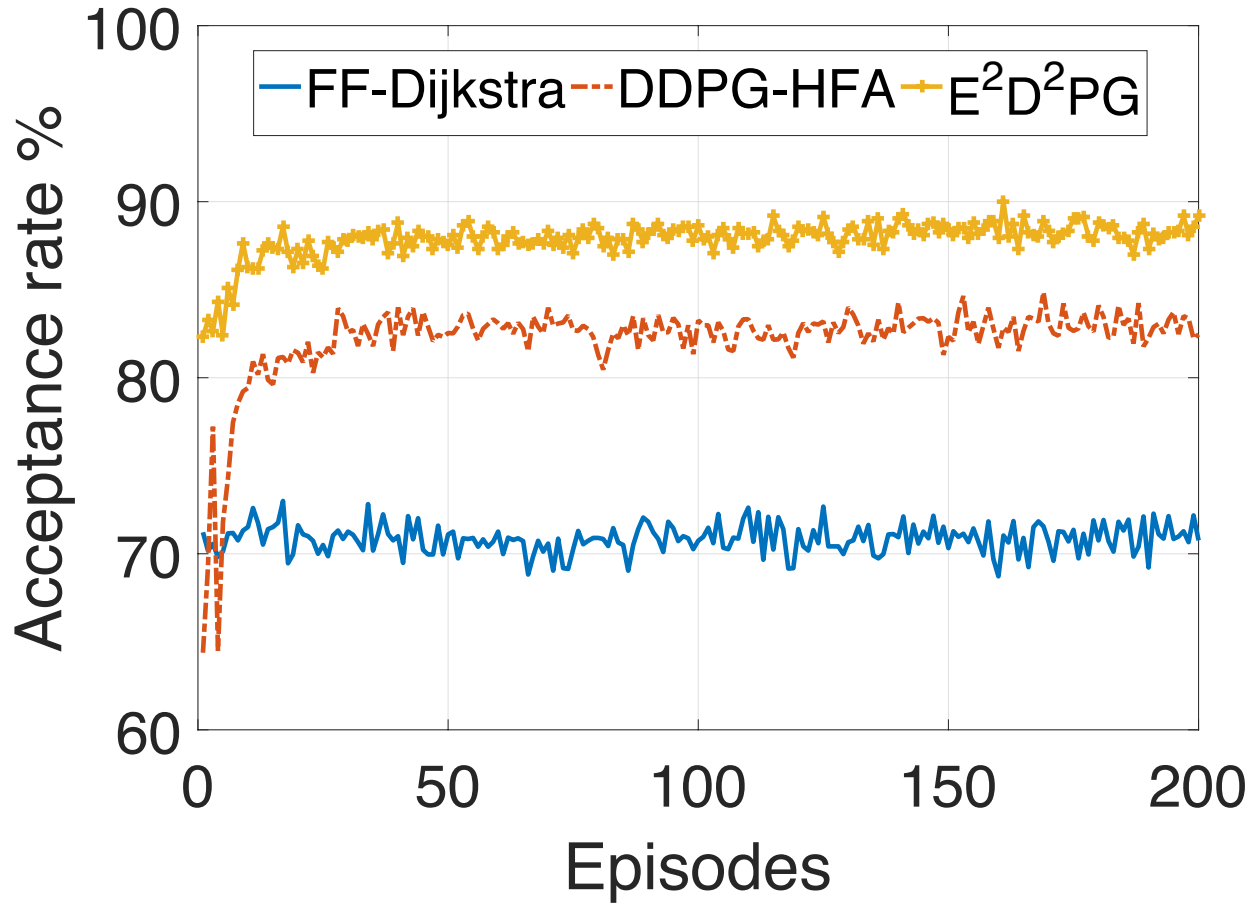


RESULTS ON THE APPLICATION OF EDDPG (1)

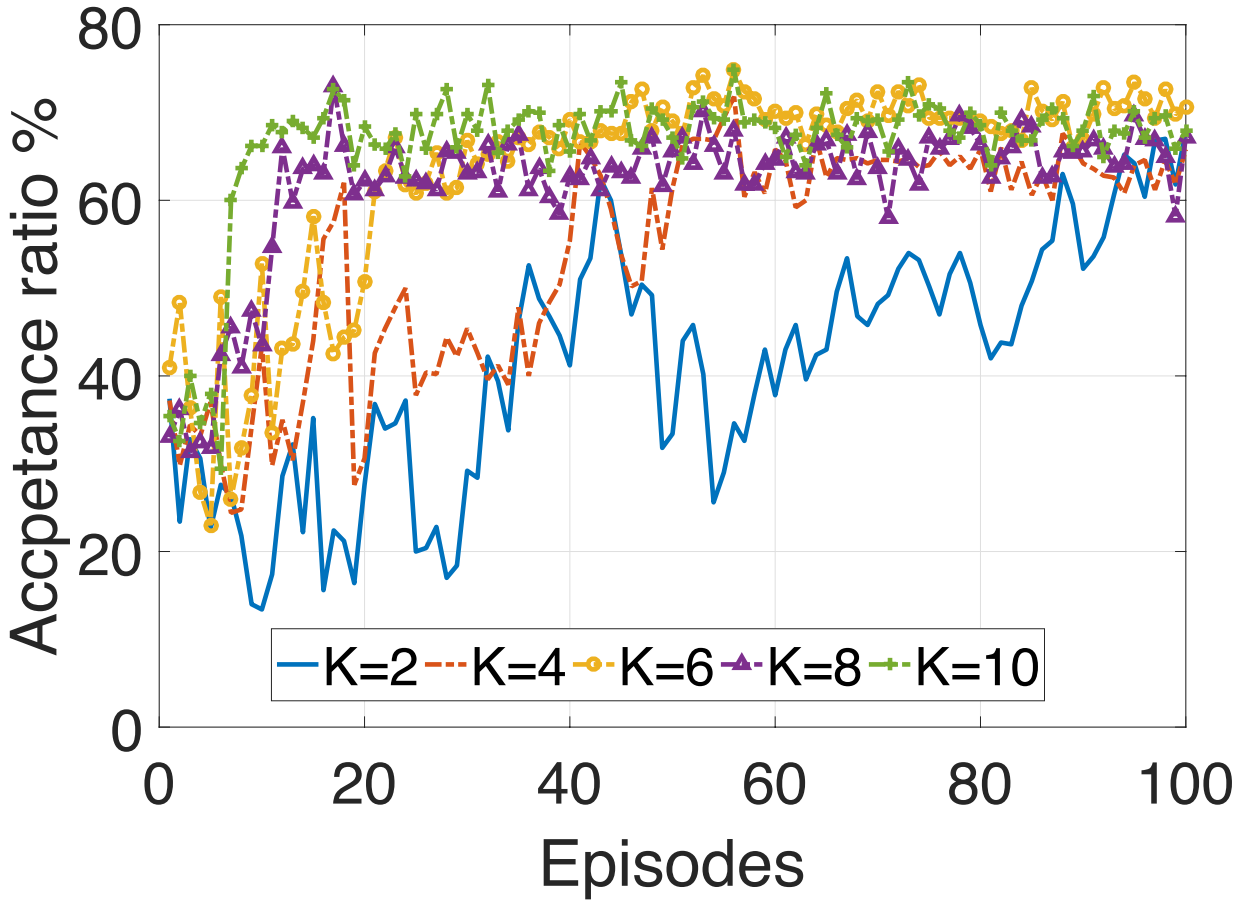




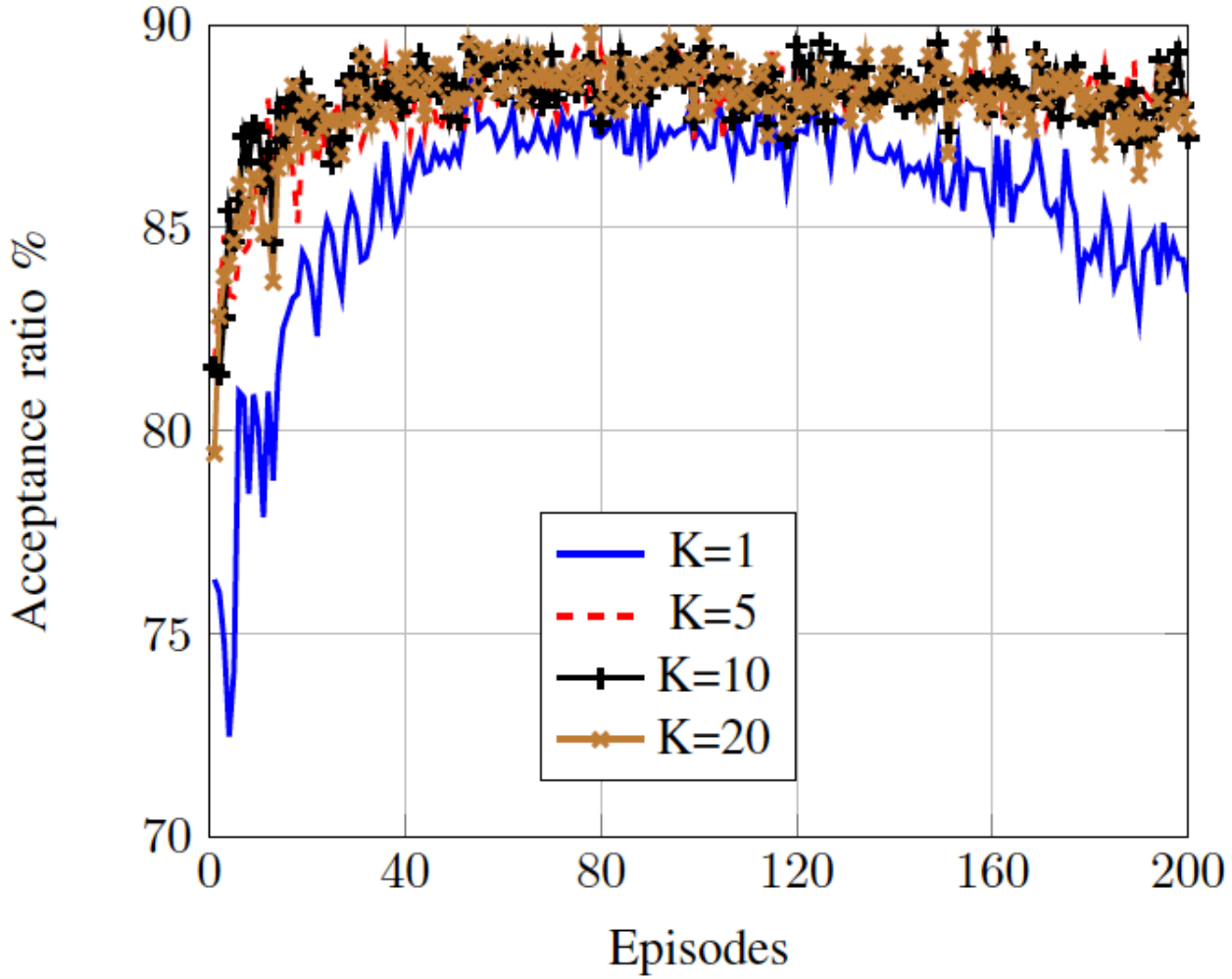
RESULTS ON THE APPLICATION OF EDDPG (2)



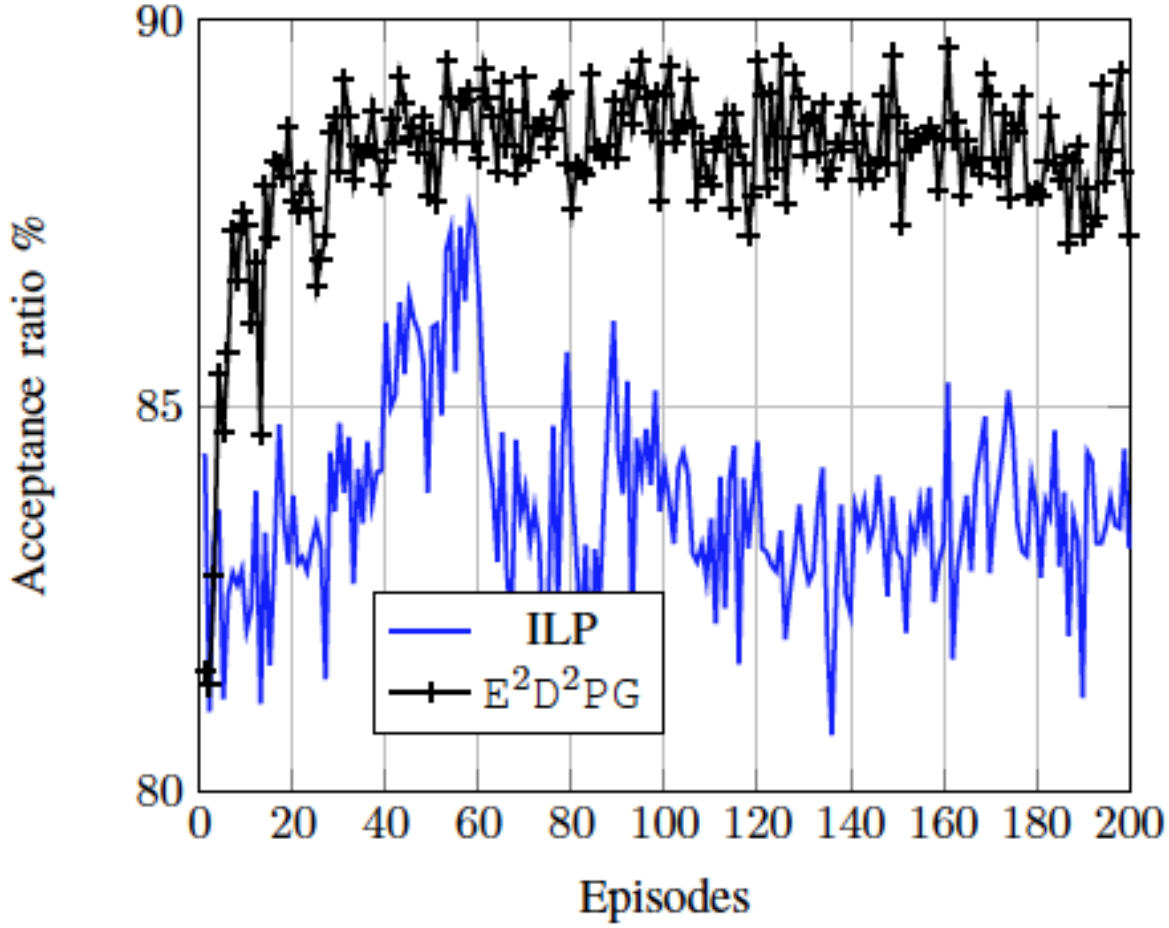
IMPACT OF THE NUMBER OF FULLY CONNECTED LAYERS



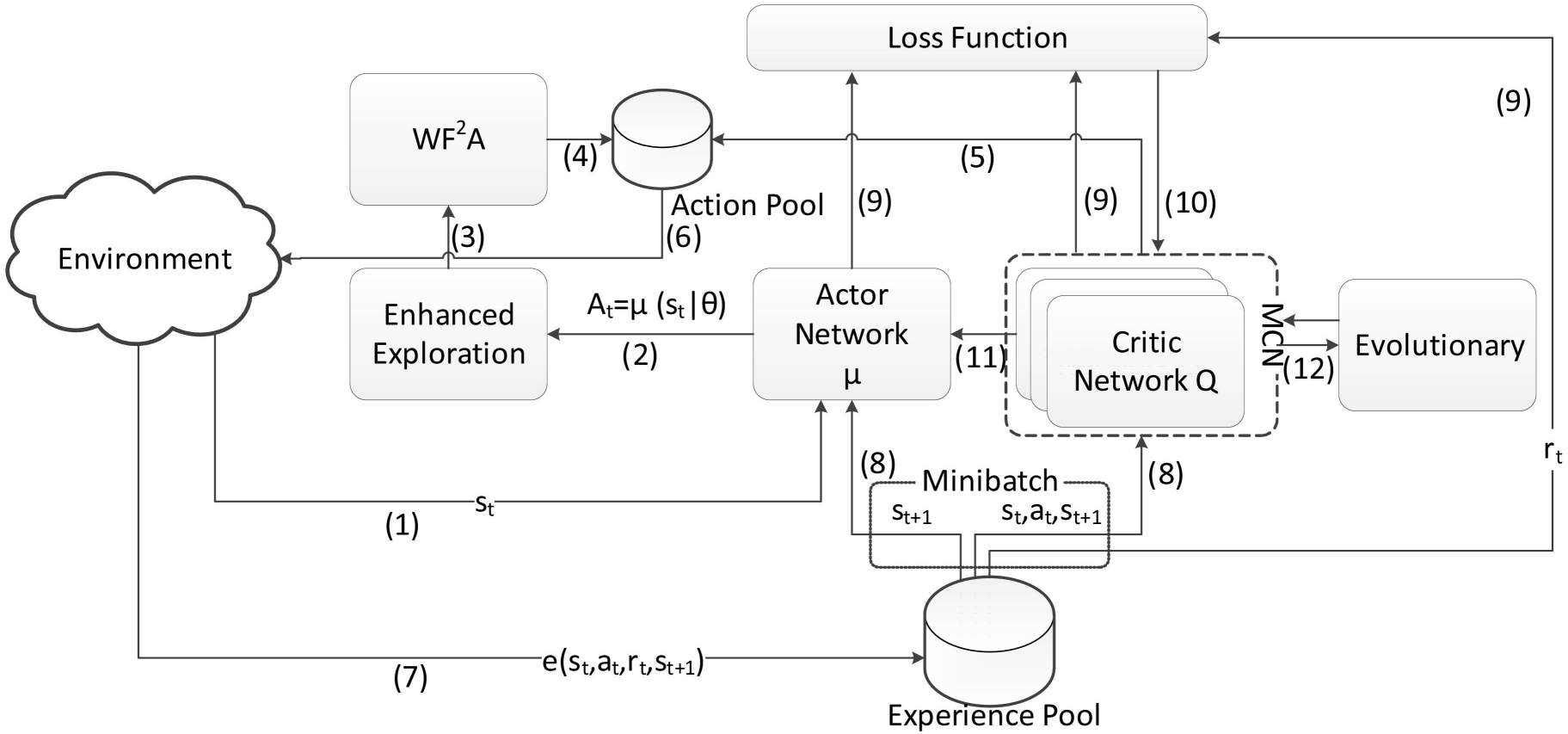
IMPACT OF THE NUMBER OF CRITIC NETS



RESULTS ON THE APPLICATION OF EDDPG (3)



EVOLUTIONARY ACTOR-MULTI-CRITIC MODEL



WEIGHTED FIRST FIT ALGORITHM (WF2A)(*)

For VNF embedding:

Step 1: Sort substrate nodes in terms of their weights

Step 2: Attempt deploying VNF at the lowest weight substrate node

If the substrate node can host the VNF

Step 3a: Update remaining resources of the substrate node

If the substrate node cannot host the VNF:

Step 3b: Remove that node from the selection process and back to step 2

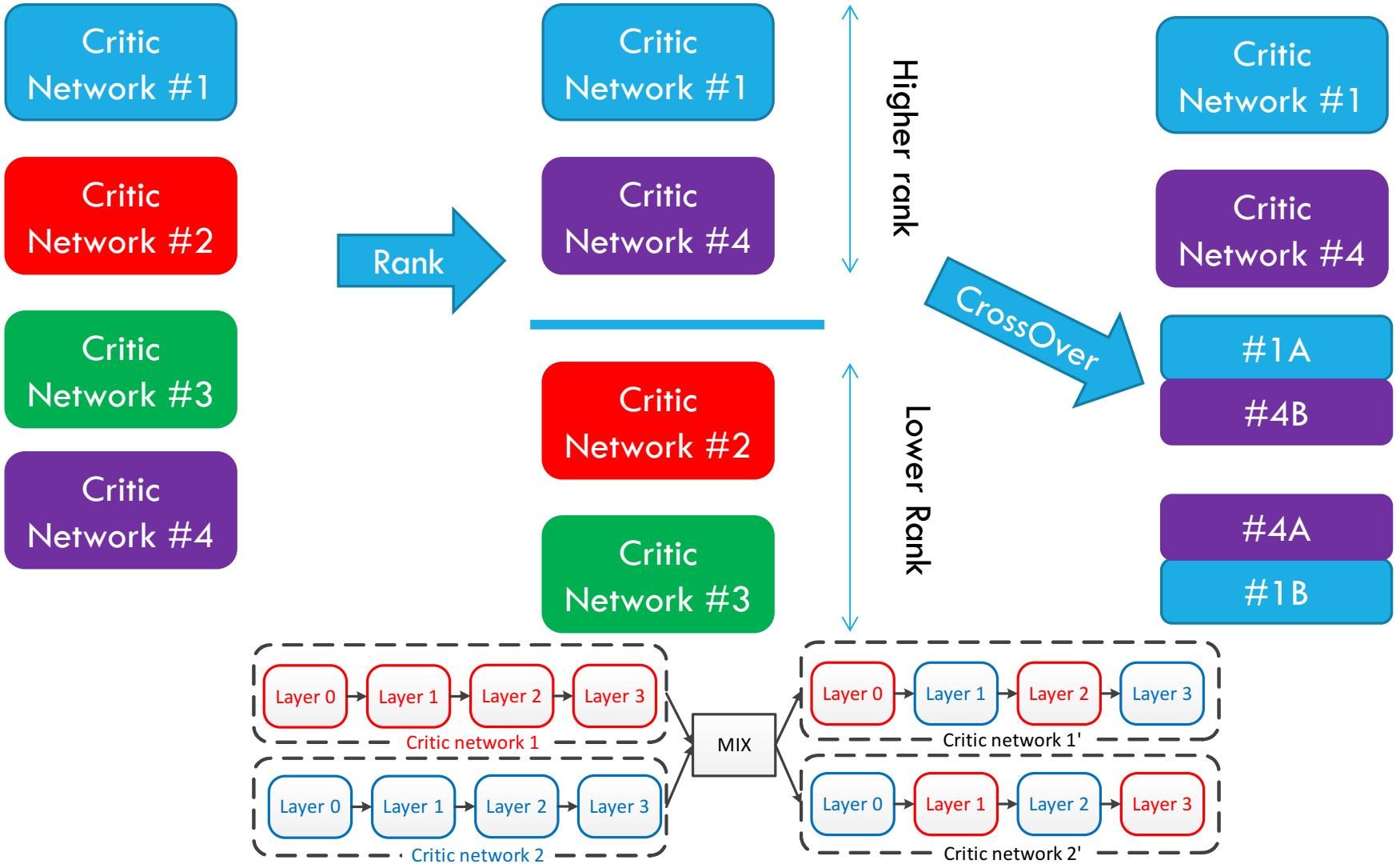
For VL embedding:

Use Dijkstra algorithm to identify the lowest cost path to connect VNFs

Final allocation decision

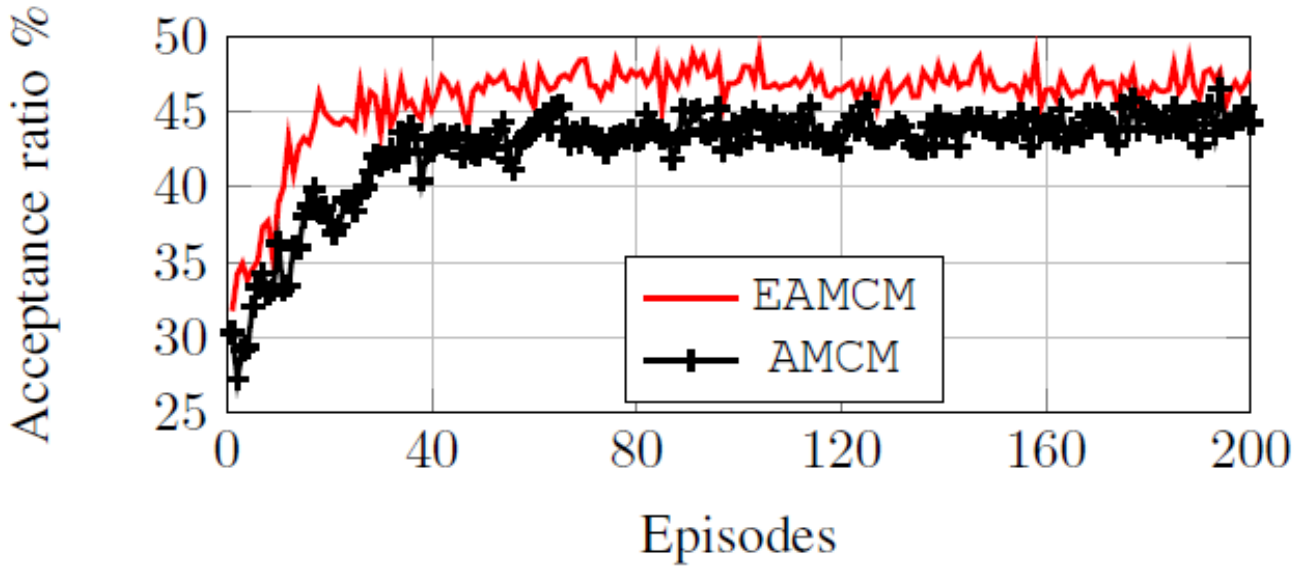
(*) P. T. A. Quang, Y. Hadjadj-Aoul and A. Outtagarts, "A Deep Reinforcement Learning Approach for VNF Forwarding Graph Embedding," in *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1318-1331, Dec. 2019.

EVOLUTIONARY ACTOR-MULTI-CRITIC MODEL



SOME RESULTS

EAMCM: Evolutionary Actor-Multi-Critic Model



ENVIRONMENT

Simulated environment

- Python, Tensorflow, Keras
- OMNET++
- REST interfaces

Emulated environment

- Python, Tensorflow, Keras
- MININET (ContainedNet - Docker)
- Orchestrator + SDN
- REST interfaces

CONCLUSIONS

DRL are very efficient in addressing the targeted issue

Considering resources variation between episodes (ongoing work)

Lack of explainability

- DNN are inherently black boxes.
- We don't know:
  - When it will work, when it will fail
  - No guarantees

Efforts are still needed …