IEEE International Conference on Communications
14-23 June 2021 // Virtual / Montreal

# DRL-SFCP: Adaptive Service Function Chains Placement with Deep Reinforcement Learning


Tianfu Wang, Qilin Fan*, Xiuhua Li, Xu Zhang, Qingyu Xiong, Shu Fu and Min Gao

Presenter

Chongqing University

Nanjing University

# Content


A Introduction on NFV

B SFCP Problem Formulation

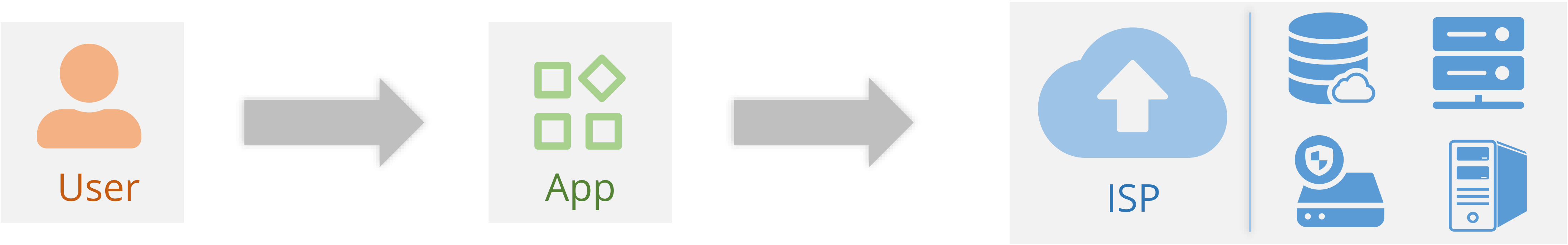
C Proposed DRL-SFCP Model

D Performance Simulation

E Conclusion and Future work

# Introduction

## Traditional Network Architecture

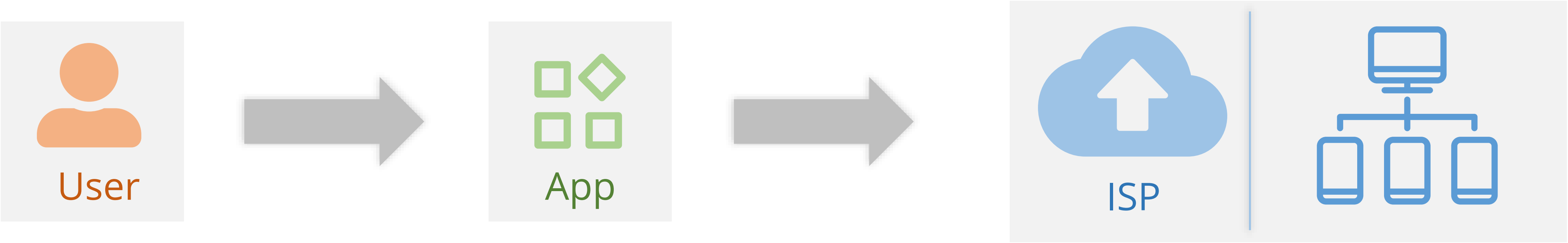



## Difficulties

- Management
- Deployment
- Consumption
- Programmability

# Introduction

## Network Function Virtualization




### Efficiency

Low-cost | High-security

Resource utilization

### Flexibility

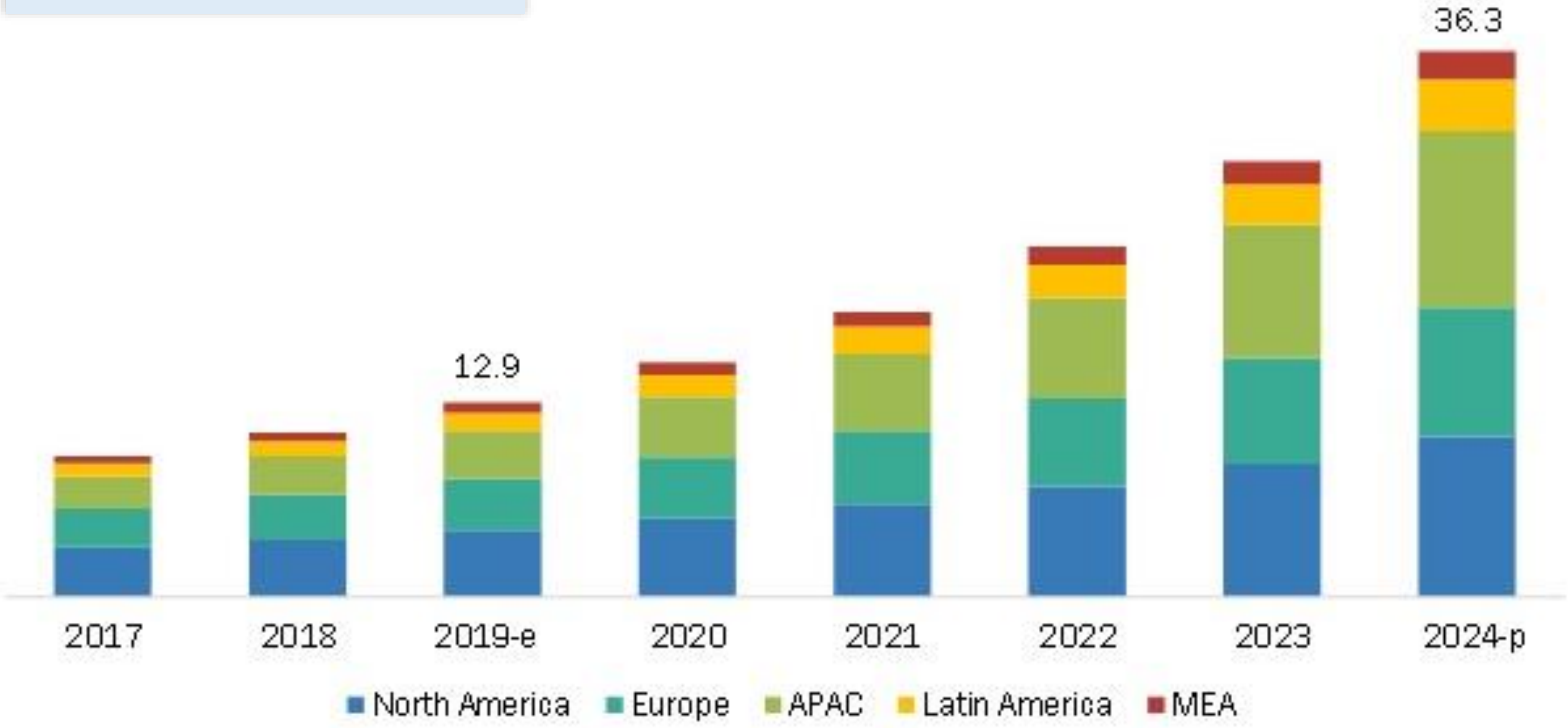
Schedulable | Programmability

Management difficulty

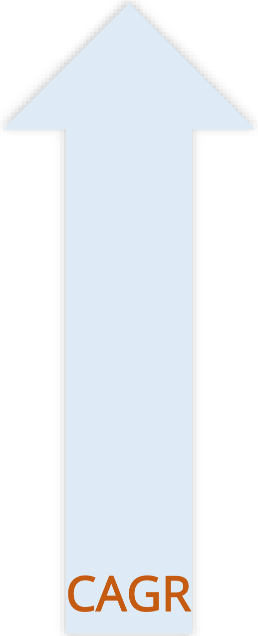
# Introduction

## NFV Market

BY REGION (USD, BILLON)




22.9%




Compound Annual Growth Rate

Major Factors

the growing need for advanced network management systems to handle the increasing network traffic and complexities.

*Image from MarketsandMarkets Research

# Introduction

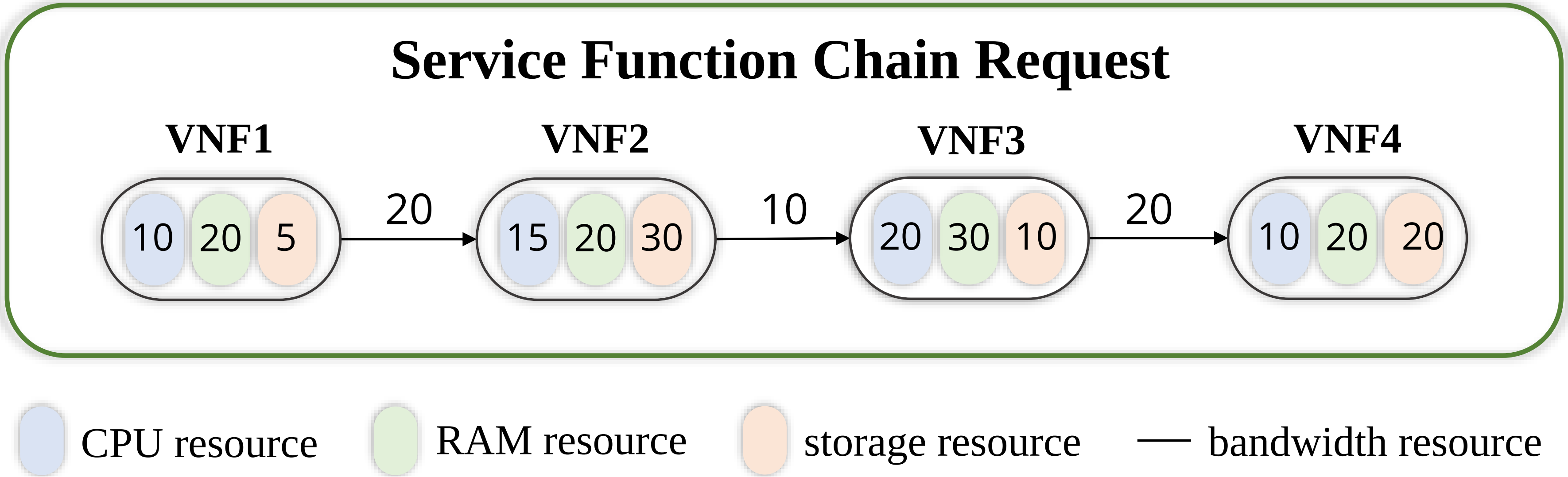
User App

Network Service

NFV framework

**SFC**

service function chain




A series of ordered VNFs (**V**irtual **N**etwork **F**unctions)

# Introduction

SFC Placement Problem

NP-hard

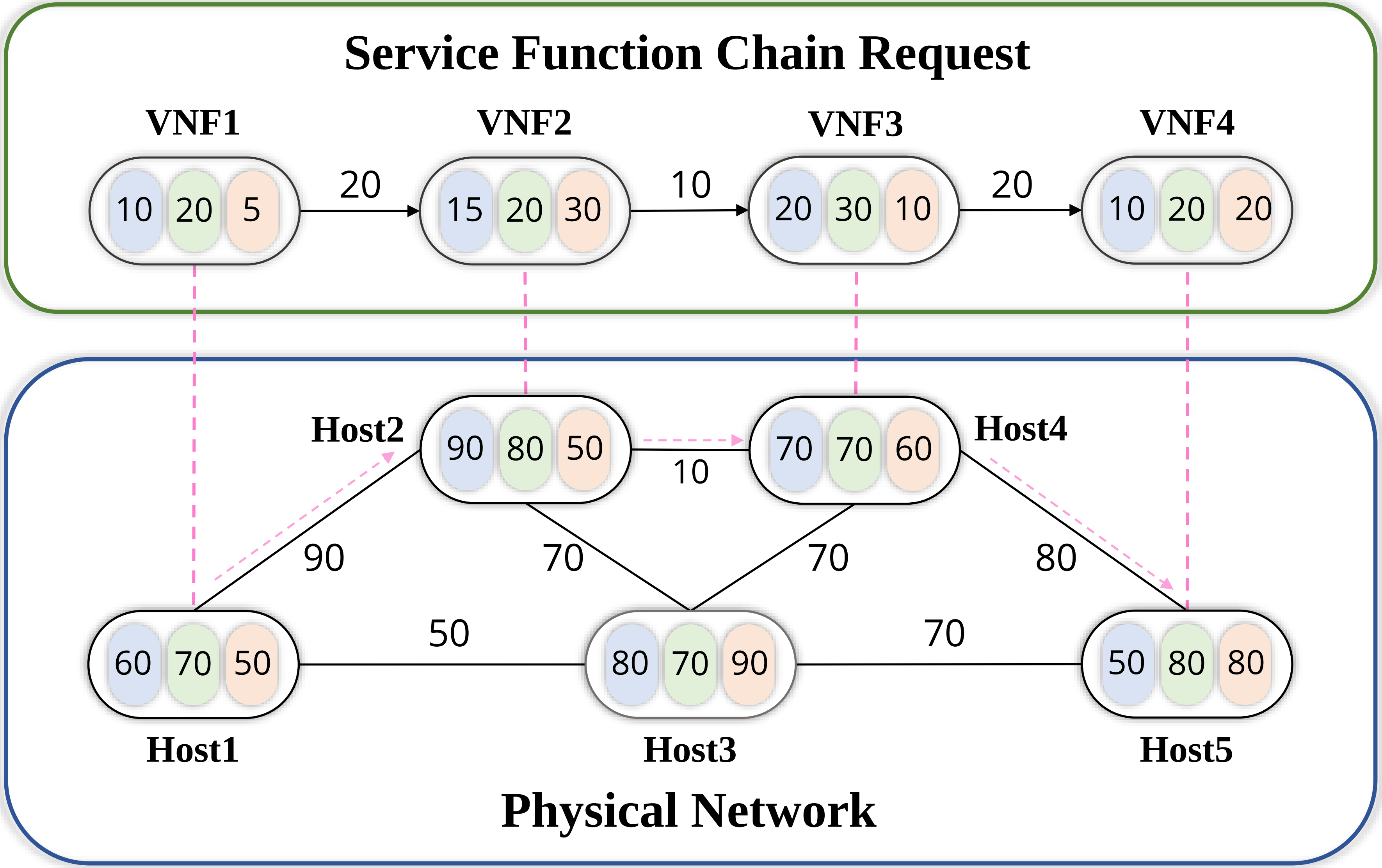
Constricted CO

CPU resource

RAM resource

storage resource

— bandwidth resource



# Introduction

## Existing Solutions

**A** Mathematical optimization-based

Require the prior knowledge of SFCs

- Integer Linear Programming
- Binary Integer Programming
- Integer Linear Programming

**B** (Meta) Heuristic-based

Fall into the local optimum and static scene

- Global Resource Control
- Node Rank based on degree
- Constructive Particle Swarm

**C** Reinforcement learning-based

Large search space & manually selected features

- Q learning-based
- Dynamic programming-based
- Policy gradient-based

# Formulation

## System

- Physical network — A weighted undirected graph — $G' = (N', L')$
- SFC request — A weighted directed graph — $G^i = (N^i, L^i)$

## Constraints

Node resources constraint: $\sum_{n^i} \phi_{n'}^{n^i} r_{n^i,k} \leq R_{n',k}^r, \forall n' \in N', \forall k \in K$

Deployment constraint: $\sum_{n'} \phi_{n'}^{n^i} \leq 1, \forall n^i \in N^i$

Link bandwidth constraint: $\sum_{l^i} \phi_{l'}^{l^i} b_{l^i} \leq B_{l'}^r, \forall l' \in L'$

Path constraint: $\sum_{l' \in I(n')} \phi_{l'}^{l^i} - \sum_{l' \in O(n')} \phi_{l'}^{l^i} = \phi_{n'}^{n_d^i} - \phi_{n'}^{n_s^i}, \forall l^i \in L^i, l' \in L'$

## Objective

Maximize the long-term average revenue

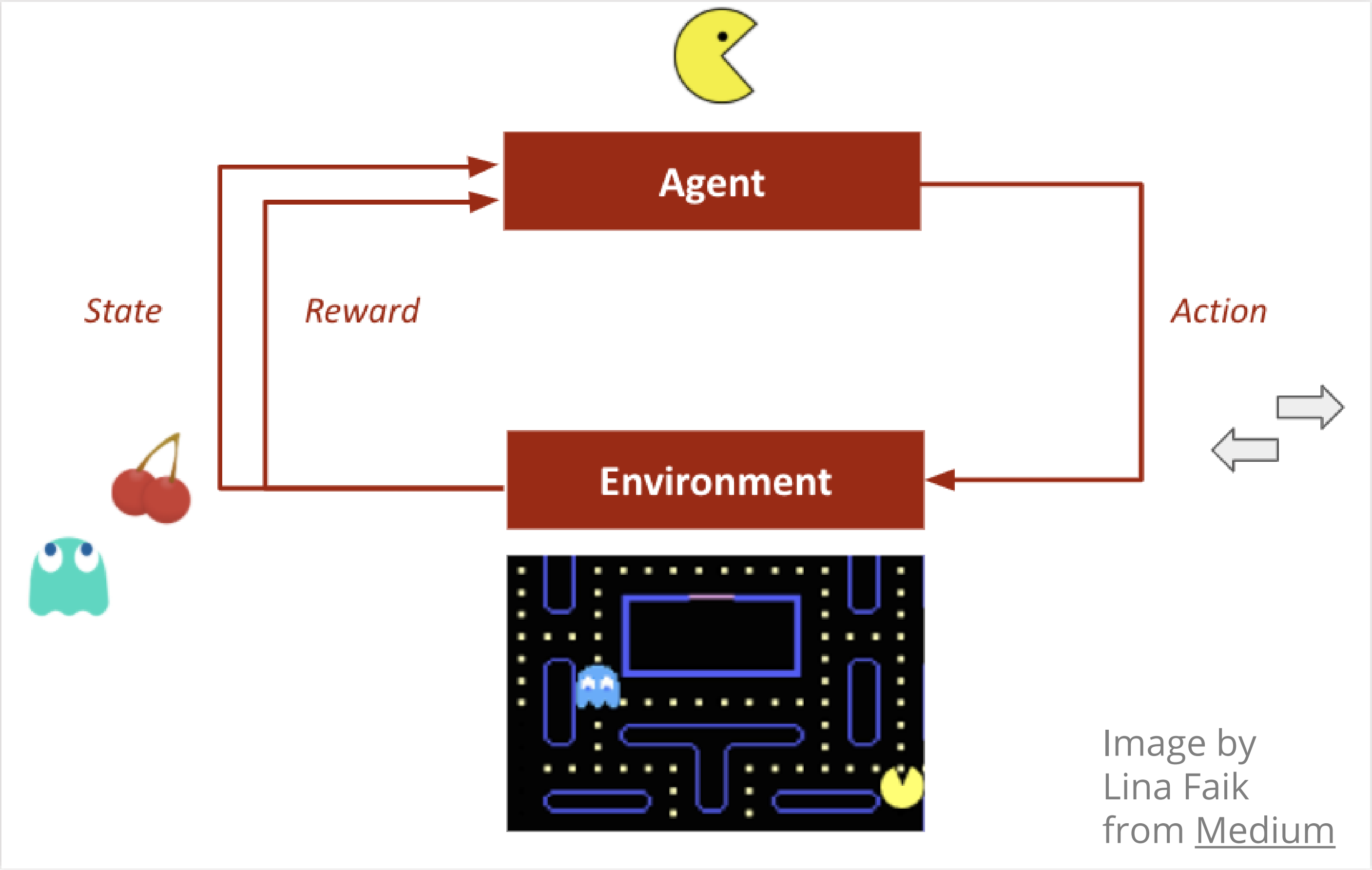
Max $R(\pi) = \lim_{\tau \to \infty} \frac{1}{\tau} \sum_{i \in I_\tau} rev(i)$

Single SFC request revenue

$$rev(i) = \begin{cases} \mu_k \sum_{n^i} r_{n^i,k} + \eta \sum_{l^i} b_{l^i}, & \text{if } i \text{ is accepted,} \\ 0, & \text{otherwise,} \end{cases}$$

# Model

## Basic RL

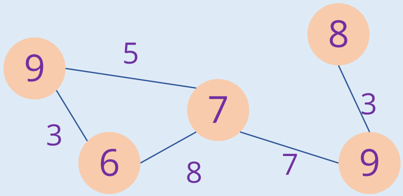



| | |
|---|---|
| Agent | make placement decisions with Neural Network |
| Env. | Physical Network<br>Serialized SFC requests |
| State | Current situation of PN<br>Demand of underway SFC |
| Action | One of physical nodes to accommodate the VNF |
| Reward | Returned to award or punish the agent's behavior |

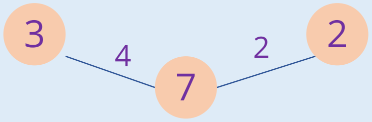
# Model

## Env

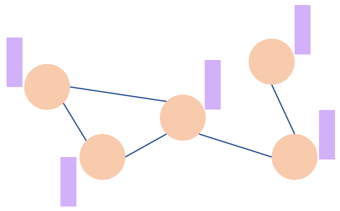
Physical Network

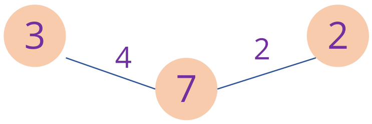



Virtual Network



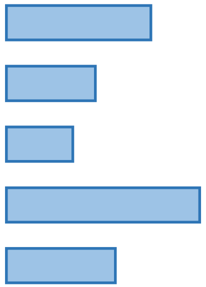

## Agent





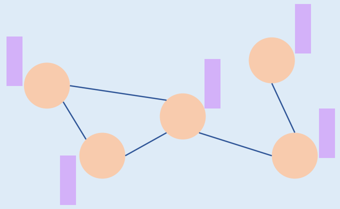

Neural Network

Action Probs



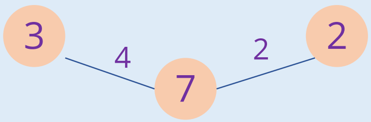
## State

Physical Network



Max Node Resource
Available Node Resource
Max Sum(Bandwidth)
Available Sum(Bandwidth)

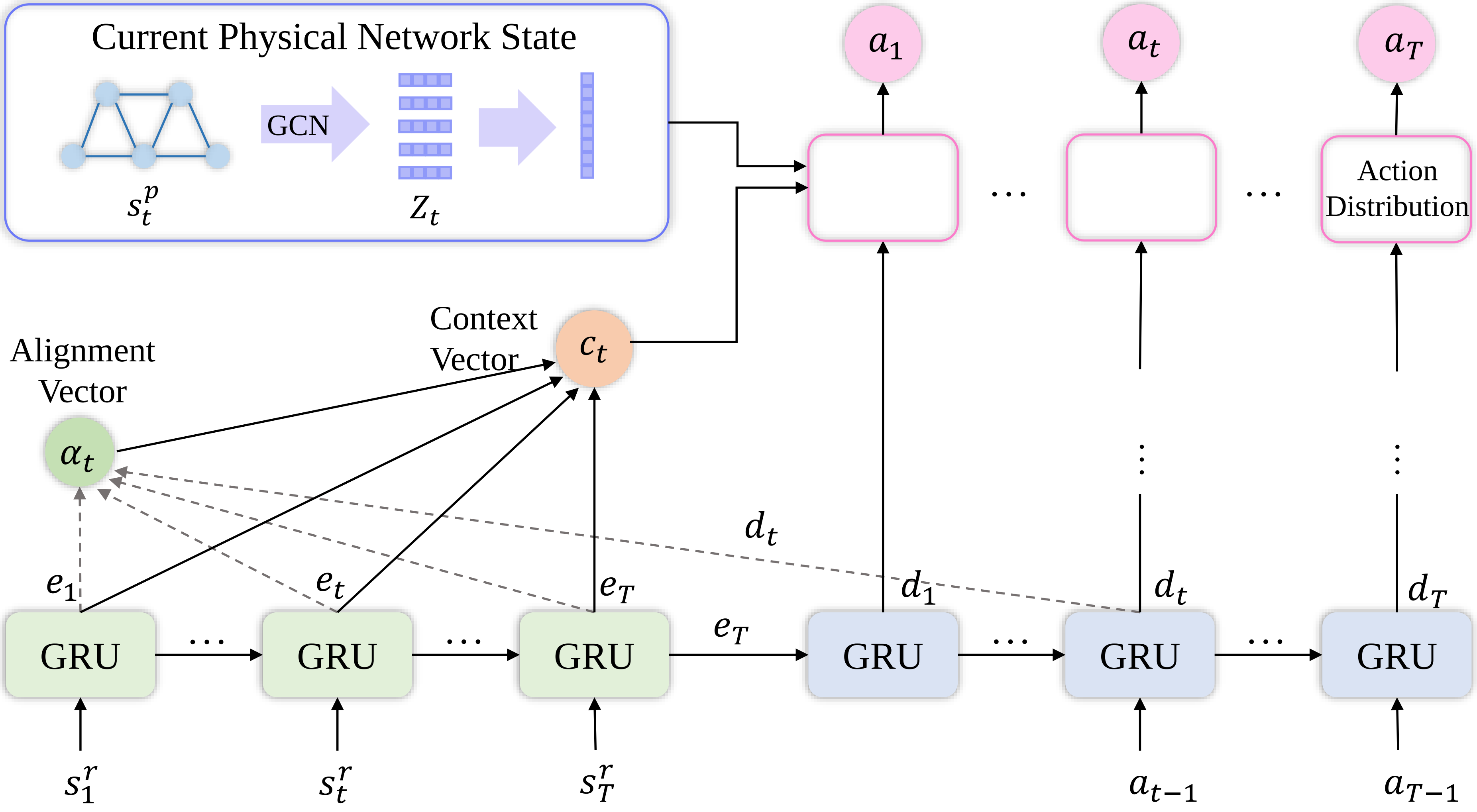
Virtual Network




Node Request
Bandwidth Request
Number of Pending Nodes

# Model

## Agent (Neural Network Architecture)




### Key Components

Embedding of physical network

Embedding of SFC request

Policy generation

# Embedding of Physical Network with GCN

$\boldsymbol{Graph} = \boldsymbol{G}(\boldsymbol{X},\boldsymbol{A})$

$\boldsymbol{X}$: Nodes Feature

$\boldsymbol{A}$: Adjacency matrix

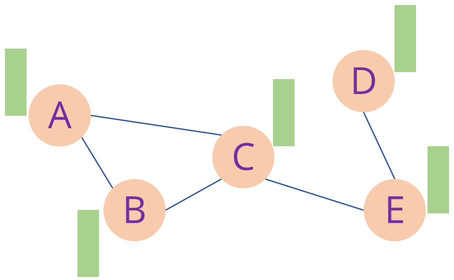
$$Z = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}XW\right)$$

$$\tilde{A} = A + I_N$$

## Input

Node Features

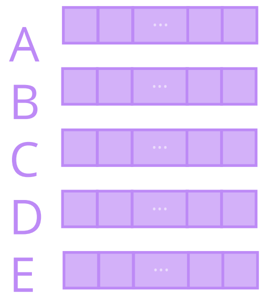
$$\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$$




## Output

Node Representation

$$\mathbf{h}' = \{\vec{h'}_1, \vec{h'}_2, \dots, \vec{h'}_N\}, \vec{h'}_i \in \mathbb{R}^{F'}$$

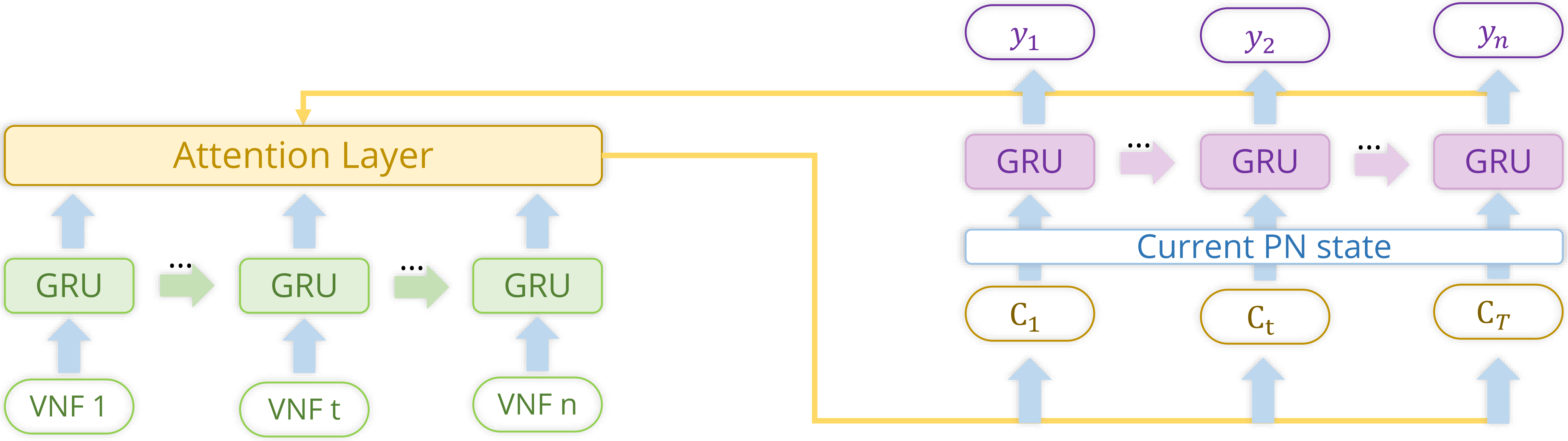



Physical Network $\xrightarrow{\text{GCN}}$ Node Representation

Thomas N. Kipf et al. “Semi-Supervised Classification with Graph Convolutional Networks”. In ICLR, 2017

# Model

## Embedding of SFC with the Encoder of Seq2Seq

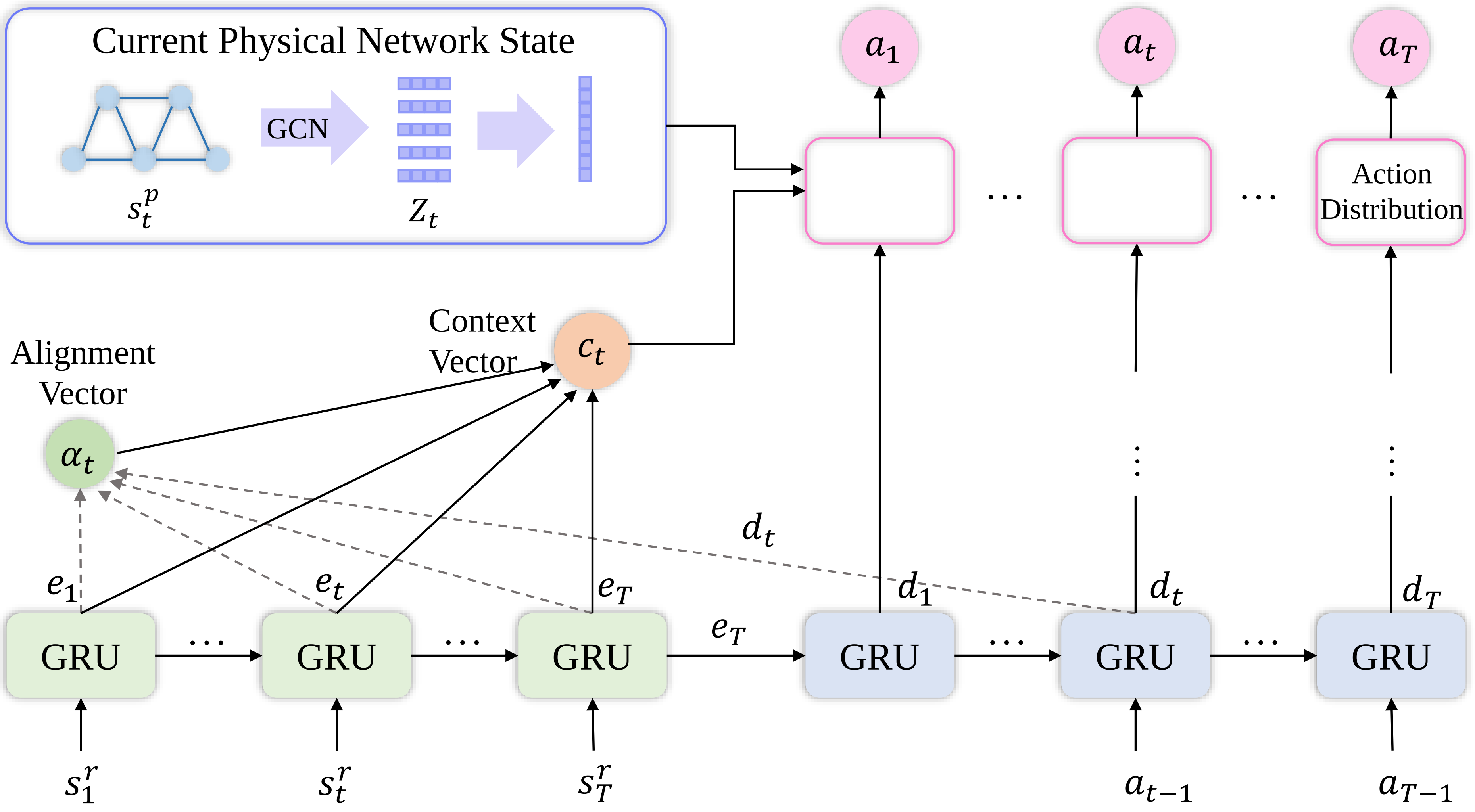



| | |
|---|---|
| Input | Resource requests of a series of VNFs in a SFC |
| Output | A series of actions (physical nodes) to accommodate VNFs |

Ilya Sutskever et al. "Sequence to Sequence Learning with Neural Networks". In NIPS, 2014

# Model

## Policy generation



# Model

## Reward Design

State → Agent → Action → **Reward**

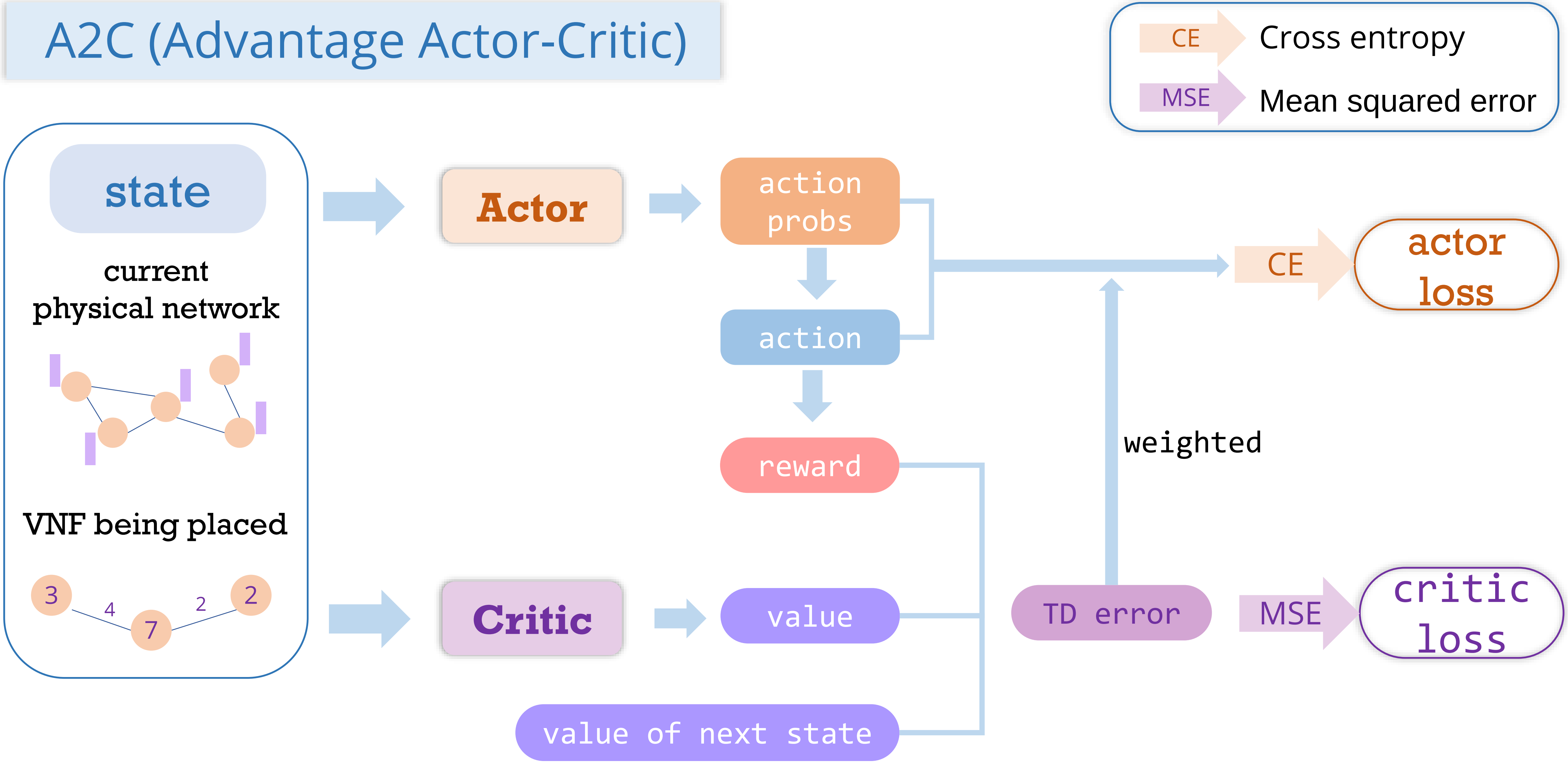
**Reward** =

- $\xi$ SFC revenue — if t != T and success (VNF is placed)
- $-\xi$ SFC revenue — if t != T and failure (SFC is rejected)
- SFC revenue — if t = T and success (SFC is accepted)

$\xi$ reward coefficient

# Model

## A2C (Advantage Actor-Critic)



# Model

## A3C (Asynchronous A2C)

Accelerate the training rate

Strengthen the model robustness

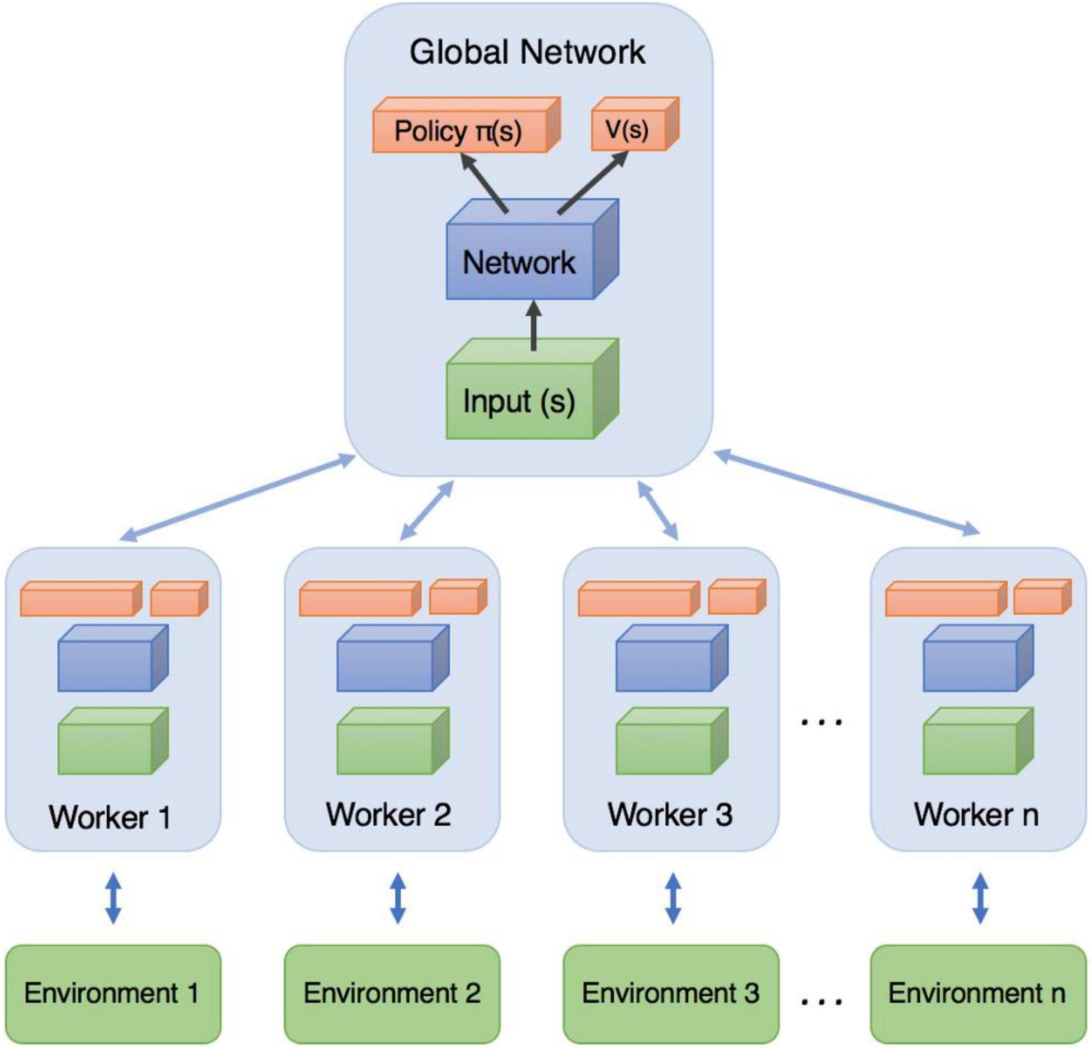



Image by
Arthur Juliani
from Medium

# Experiment

## Experimental Setup

### Model Parameters

| Name | Value | Description |
|---|---|---|
| $U$ | 4 | the number of actor networks |
| $\mu_k$ | 0.001 | the unit price of resource $k$ |
| $\eta$ | 0.001 | the unit price of bandwidth |
| $\varepsilon_\theta$ | 0.00025 | the learning rate of actor |
| $\varepsilon_\omega$ | 0.0005 | the learning rate of critic |
| $\gamma$ | 0.95 | the discount factor of TD error |
| $\xi$ | 0.125 | the reward coefficient |
| $B$ | 64 | the batch size |
| $U_{gcn}$, $U_{emd}$, $U_{enc}$, $U_{dec}$ | 64 | the units number of GCN layer, embbeding layer, encoder hidden states and decoder hidden states |

## Physical Network

**topology** About 500 Nodes and 200 Links

**Resources** Uniform distribution [50, 100]

## SFC Request

**SFC lifetime** Exponential distribution with an average of 400

**SFC length*** Uniform distribution [2, 15]

**Average arriving rate *** 20 per 100 time units

## Compared Algorithms

- GRC based on global resource capacity (L. Gong et al. INFOCOM 2014)
- MCTS using Monte Carlo tree search (S. Haeri et al. IEEE Trans Cybern 2018)

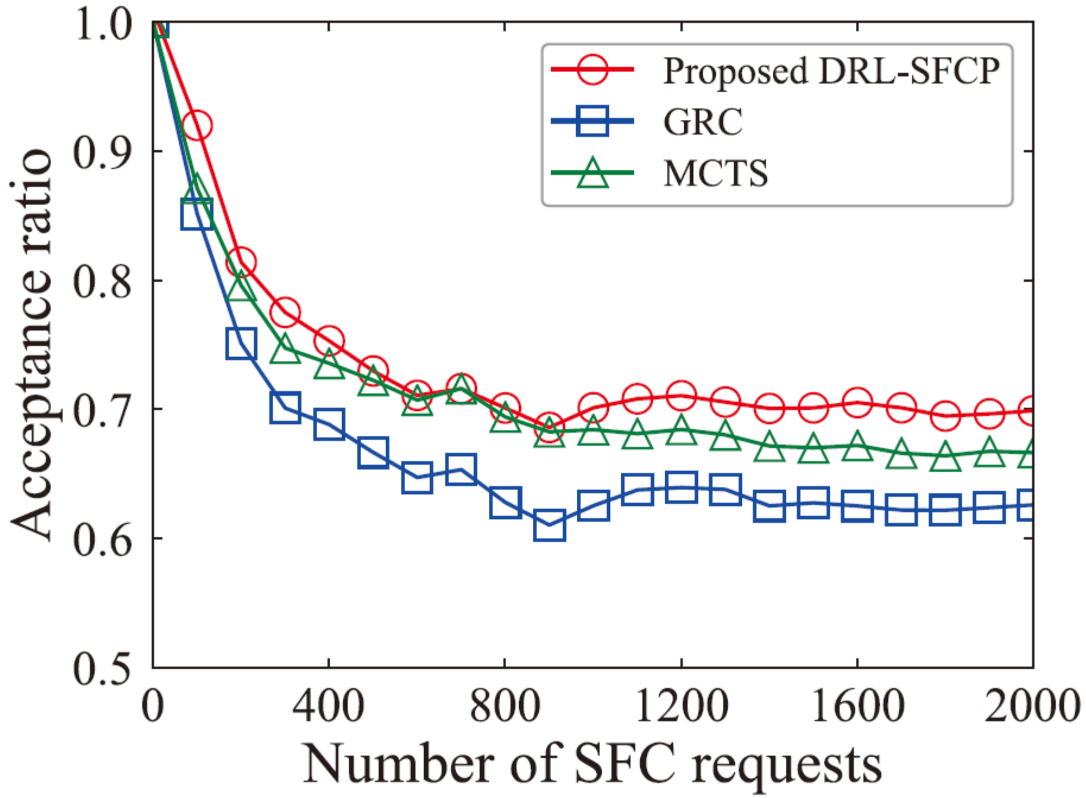
* means these settings may vary in the experiments for diverse evaluation

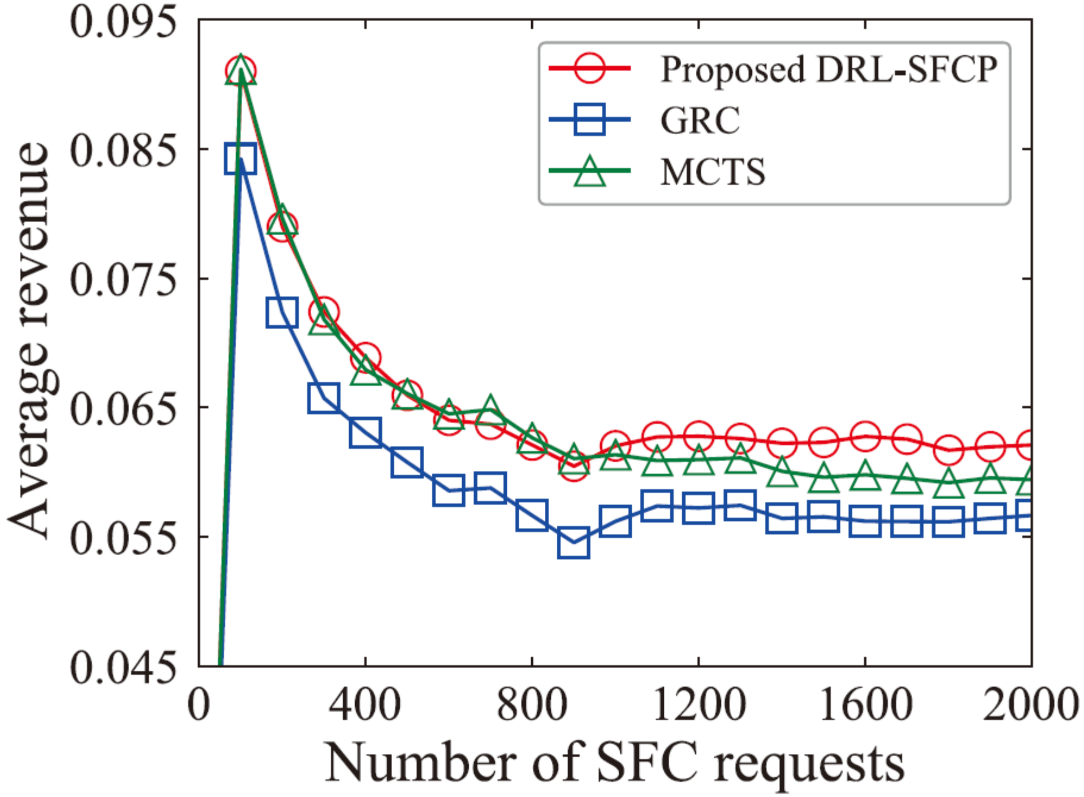
# Experiment

## Acceptance Ratio & Average Revenue

**Result** DRL-SFCP achieves greater effect on two indicators

**Ascribe** the abundant information extracted from SFC and PN



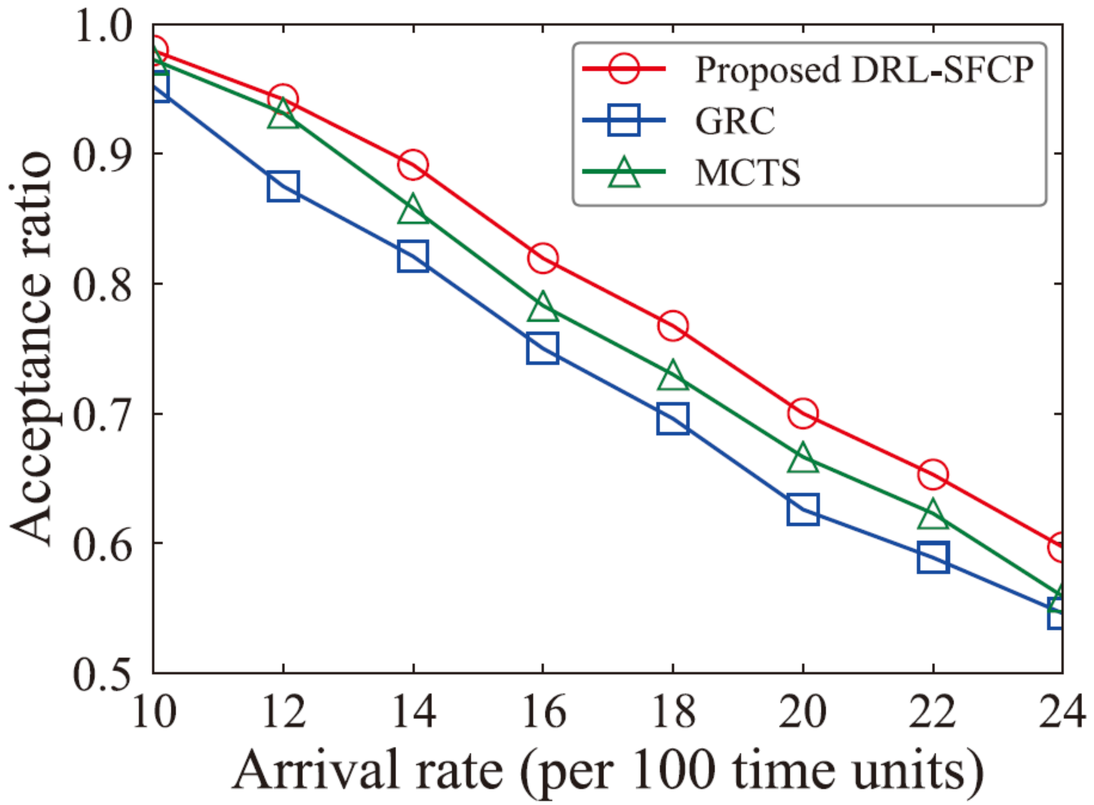



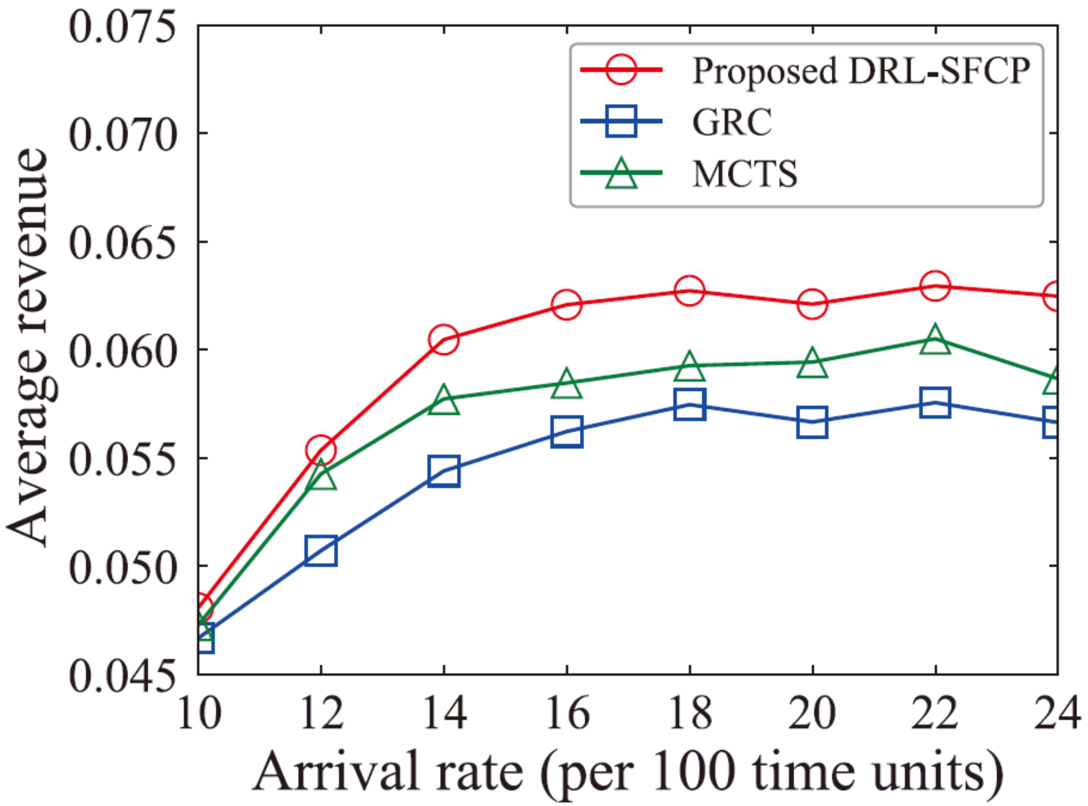
# Experiment

## In various arrival rate conditions

Result — The performance of DRL-SFCP outperforms GRC and MCTS

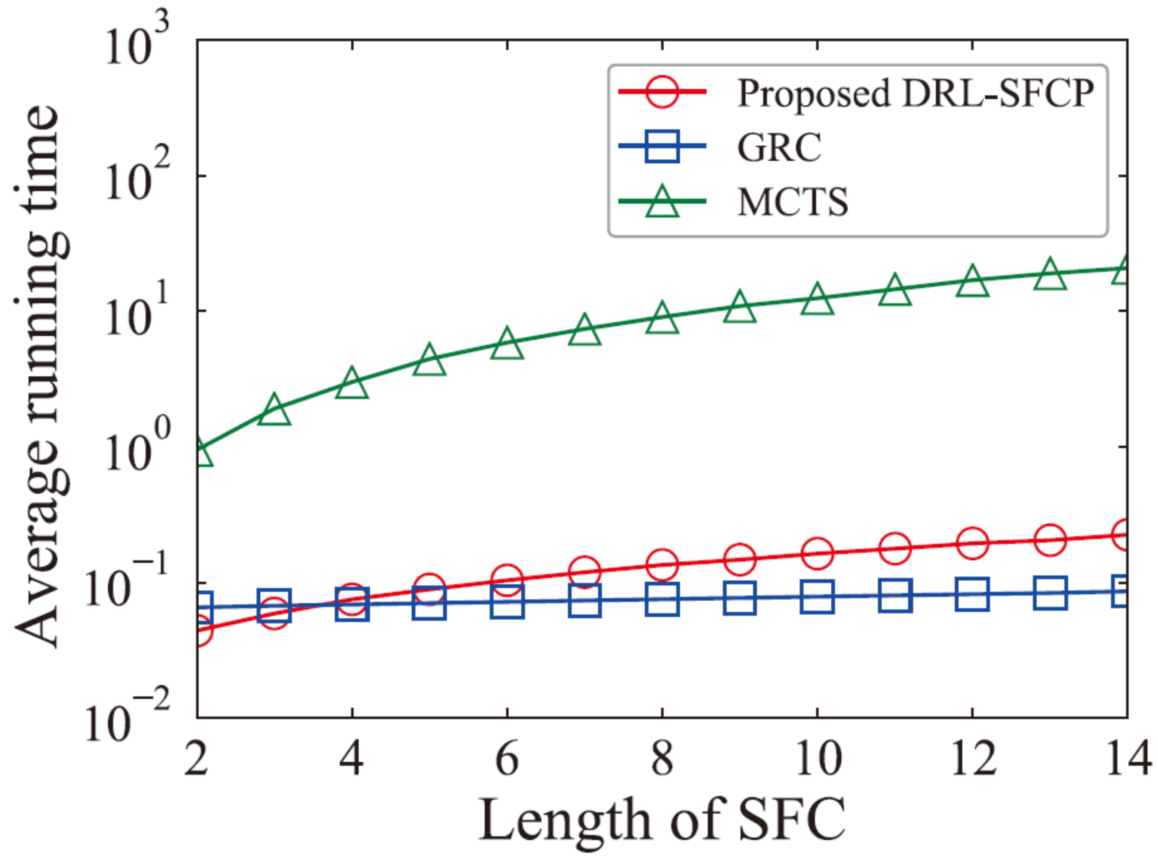
Ascribe — the excellent abilities of fitting and generalization of DNN

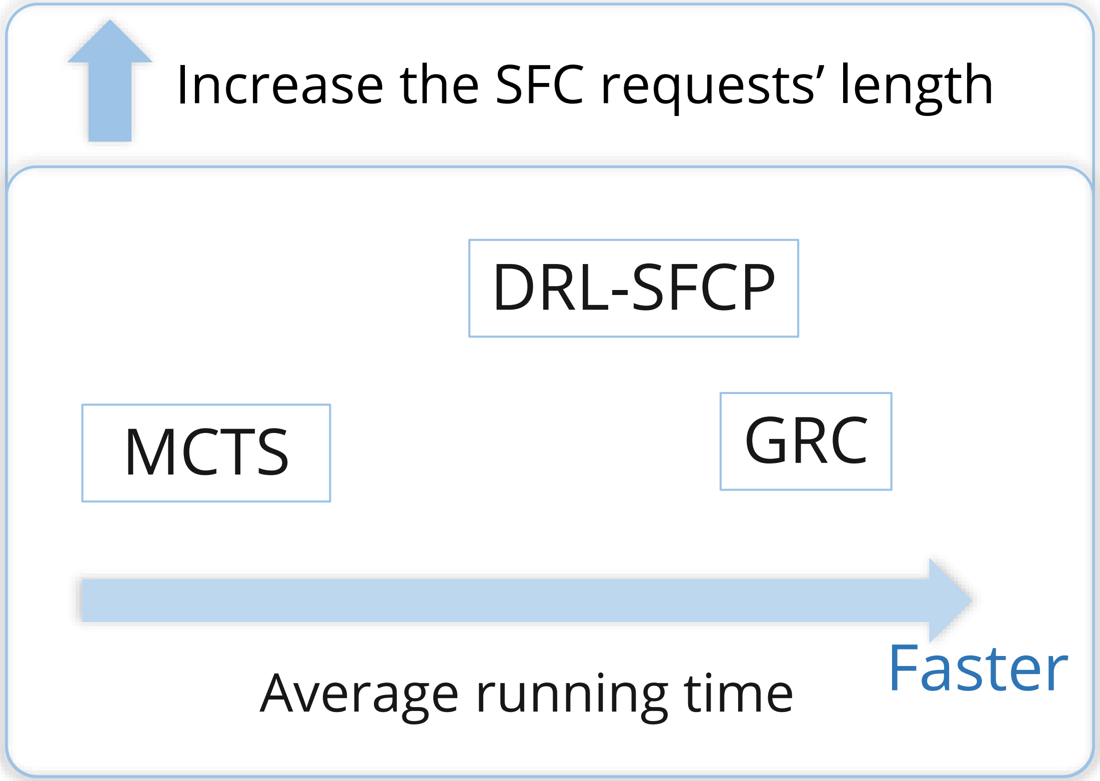


# Experiment

## Average running time




It demonstrates that
DRL-SFCP is well-suited for Application on online scenarios

# Conclusion

## Our contribution

- Adaptive DRL framework ▶ Guide online placement decision for SFC requests
- Effective NN architecture ▶ Extract the sufficient information from input features
- Parallel Training Method ▶ Enhance the training efficiency and model robustness

## Future work

**More powerful NN architectures**
i.e. GNN, Transformer

**More efficient DRL methods**
i.e. Multi-agent

**More realistic modeling scenarios**
i.e. latency, multi-flow

# THANKS

wtfly2018@gmail.com
github.com/GeminiLight/drl-sfcp

**DRL-SFCP**: Adaptive Service Function Chains Placement with Deep Reinforcement Learning

Tianfu Wang, Qilin Fan*, Xiuhua Li, Xu Zhang, Qingyu Xiong, Shu Fu and Min Gao