A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization

Giorgio Stampa*, Marta Arias, David Sanchez-Charles, Victor Muntes-Mulero, **Albert Cabellos**

acabello@ac.upc.edu

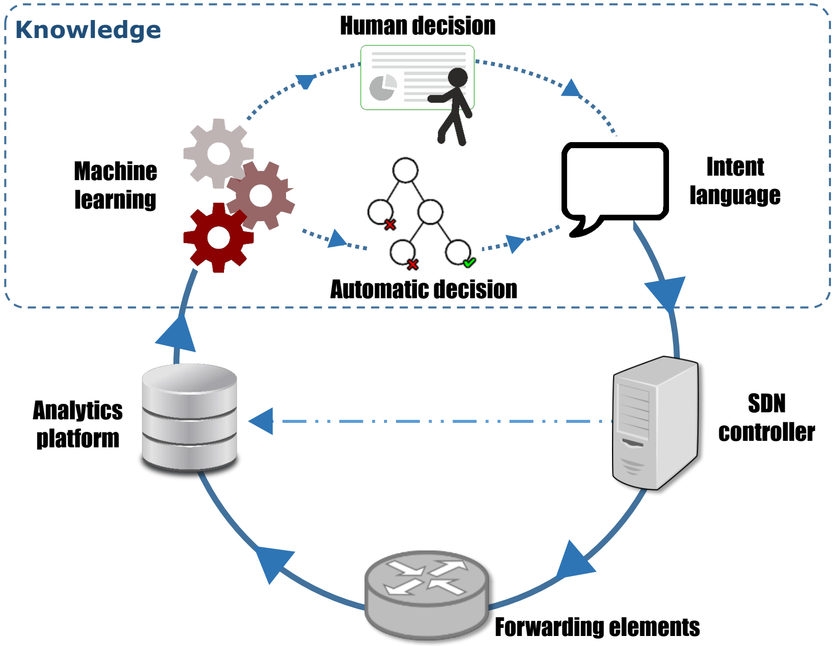
IETF 100 – NMG RG

Singapore, November 2017

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

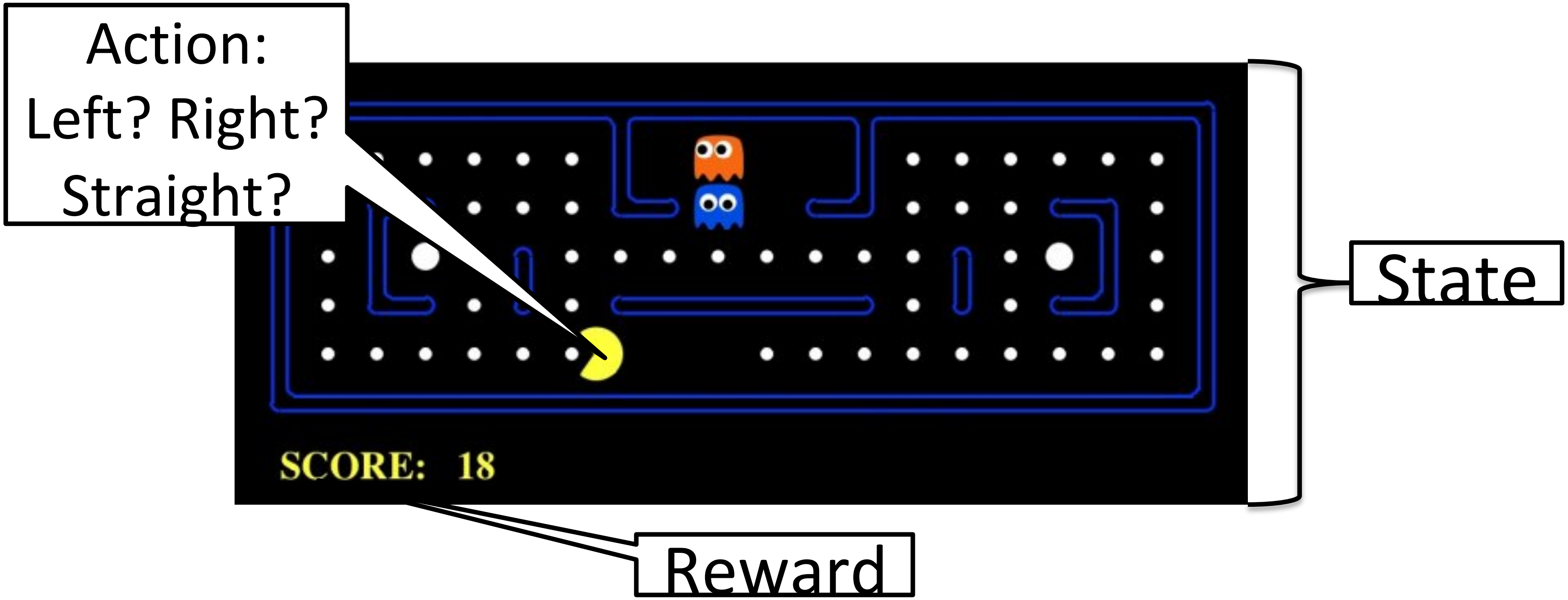
Knowledge-Defined Networking

- A Knowledge plane, **on top of control and management planes**, should allow for:
  - automation & optimization
  - prediction
- Machine Learning can take advantage of:
  - The **full view** provided by the Network Analytics platform
  - The **full control** provided by the (logically) centralized management and control planes
- **Which Machine Learning technique?**
- **How we apply it?**



A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés, D. Meyer, S. Barkai, M. J. Hibbett, G. Estrada, K. Maruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-Defined Networking," SIGCOMM Comput. Commun. 2017 https://arxiv.org/abs/1606.06222

Deep Reinforcement Learning



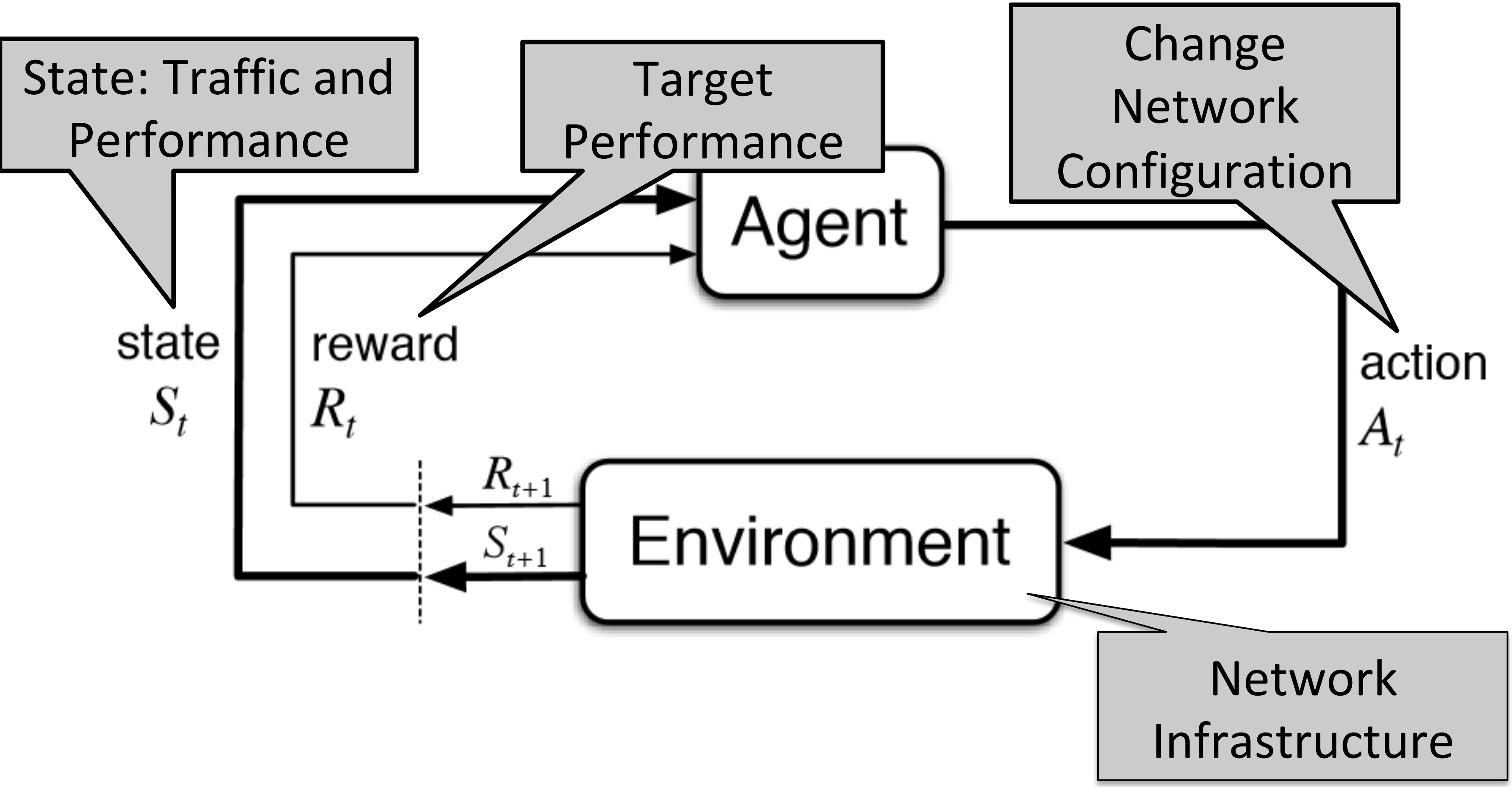
Action:
Left? Right?
Straight?

State

Reward

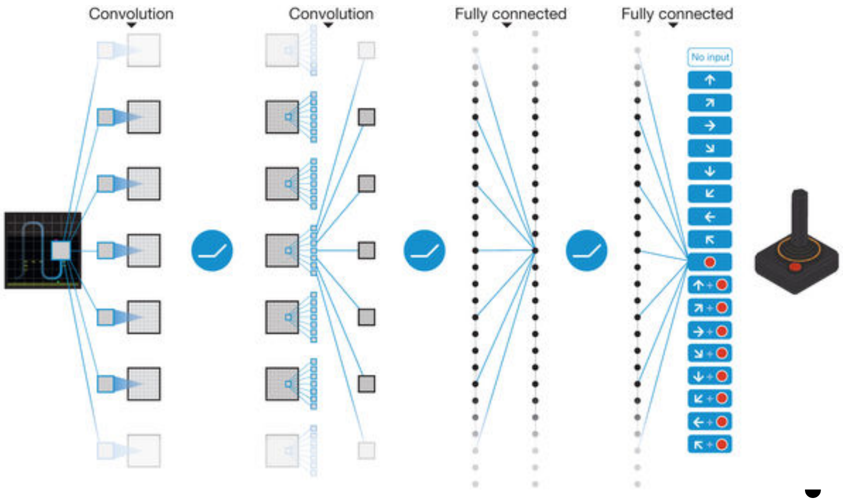
Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.

Deep Reinforcement Learning



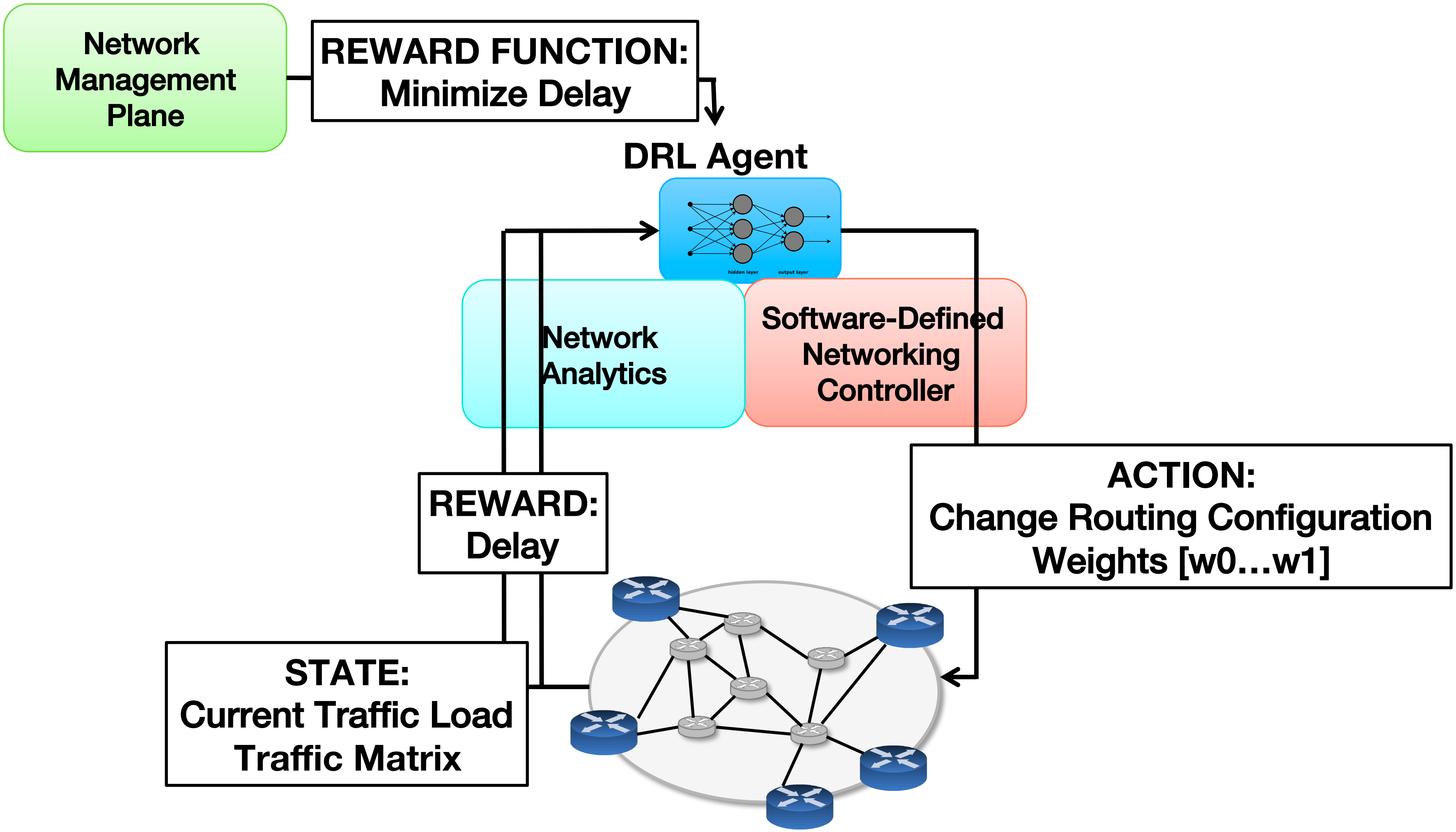
DRL: Internal Architecture

- DRL = Reinforcement Leraning + Deep Learning
- Novel Actor/Critic Architecture
    - The Actor acts upon the system
    - The Critic receives the reward function and modifies the weights of the Actor's Neural Network.
- Exploration vs. Exploitation
    - Exploration: Training on the system
    - Exploitation: Optimization of the system



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.

DRL for SDN Routing



Network Management Plane

REWARD FUNCTION:
Minimize Delay

DRL Agent

Network Analytics

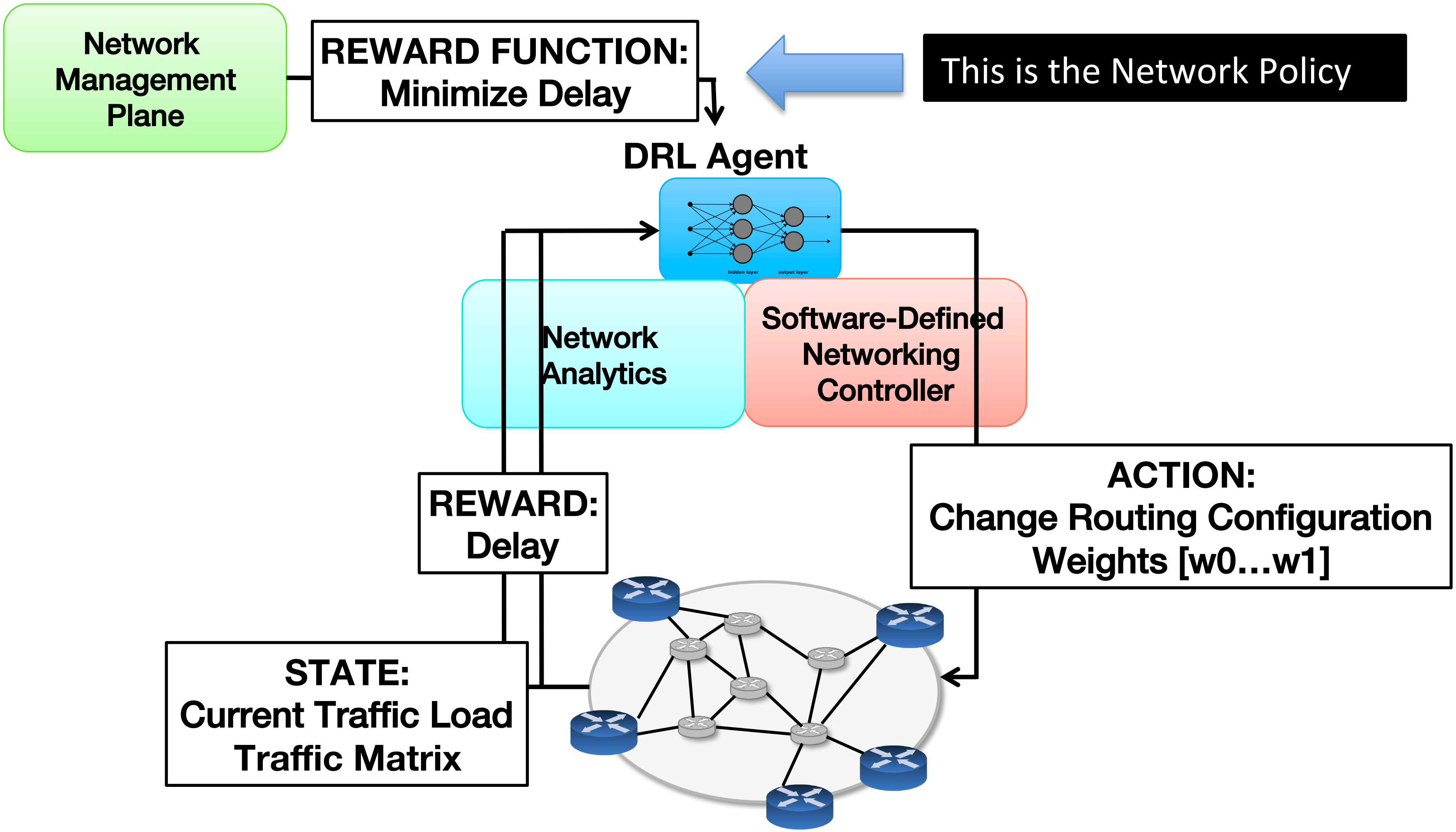
Software-Defined Networking Controller

REWARD:
Delay

ACTION:
Change Routing Configuration Weights [w0…w1]

STATE:
Current Traffic Load Traffic Matrix

DRL for SDN Routing



Network Management Plane

REWARD FUNCTION:
Minimize Delay

This is the Network Policy

DRL Agent

Network Analytics

Software-Defined Networking Controller

REWARD:
Delay

ACTION:
Change Routing Configuration Weights [w0…w1]

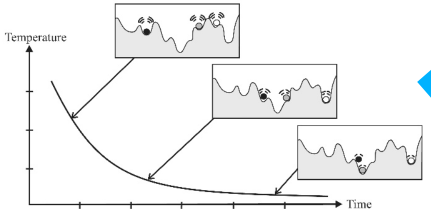
STATE:
Current Traffic Load
Traffic Matrix

Methodology

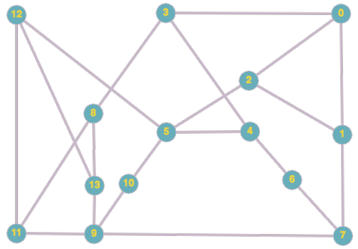
Large - Realistic topology of 79 nodes and over 200 links

1000 TMs, 1000 steps per TM (1M steps overall)

Optimization Algorithm Simulated Annealing

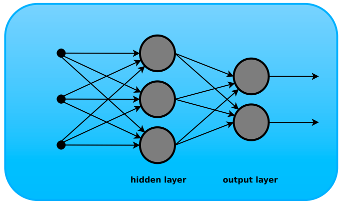


Small - Realistic topology of 13 nodes and over 25 links



Training, 100k steps of random TMs (100k steps overall)

DRL Agent

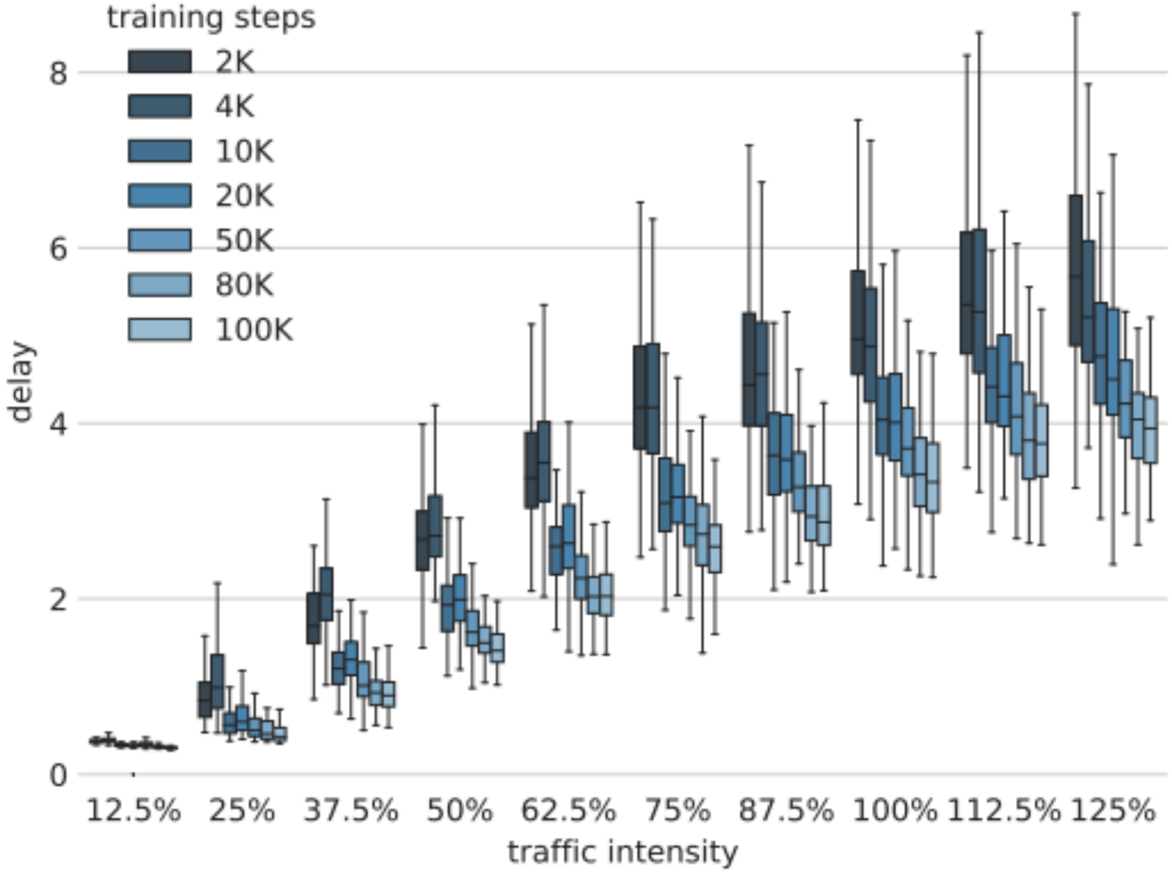


Compare Performance

Simulation: Omnet++

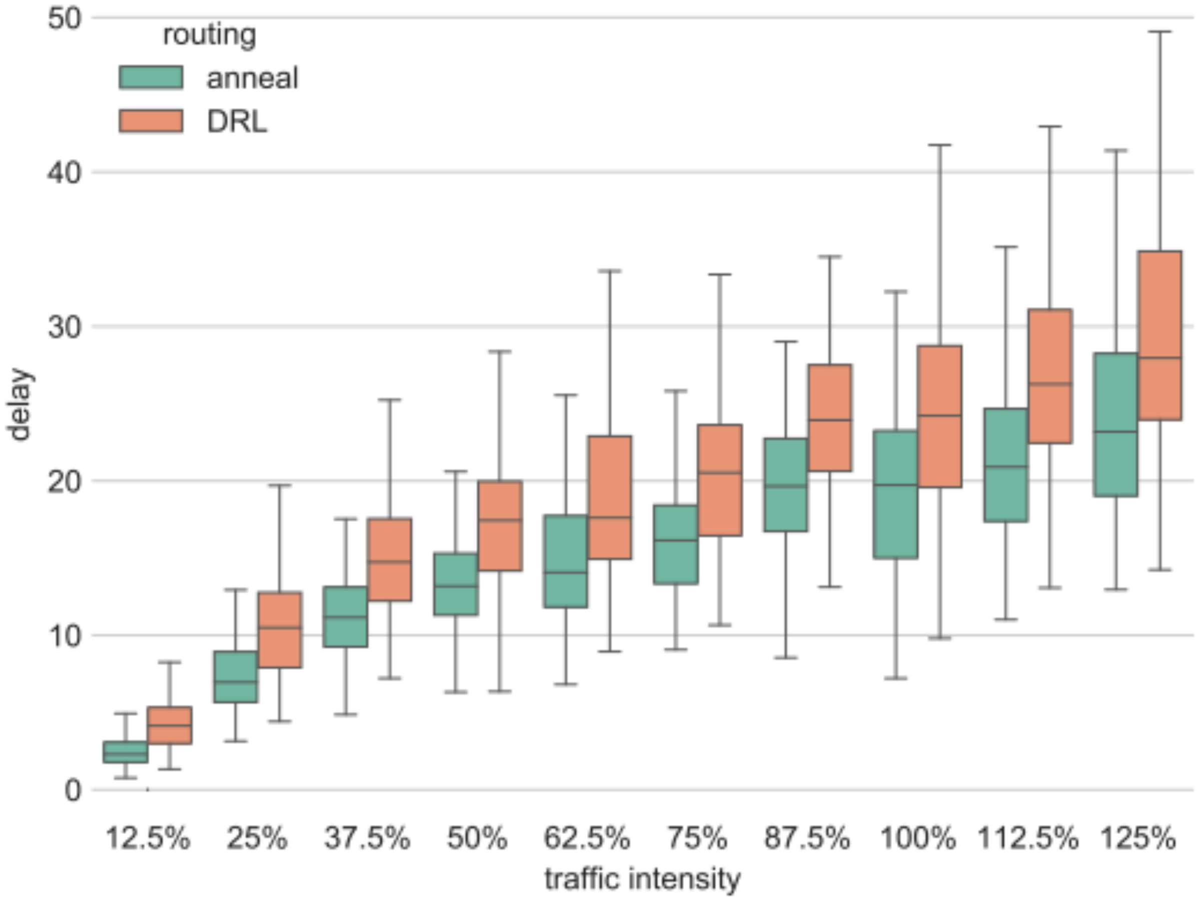
B. Quoitin et al., “IGen: Generation of router-level Internet topologies through network design heuristics,” in ITC, 2009.

Results: Learning Rate



G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntes-Mulero, and A. Cabellos,
"A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization," arXiv preprint arXiv:1709.07080, 2017.

Results: Performance



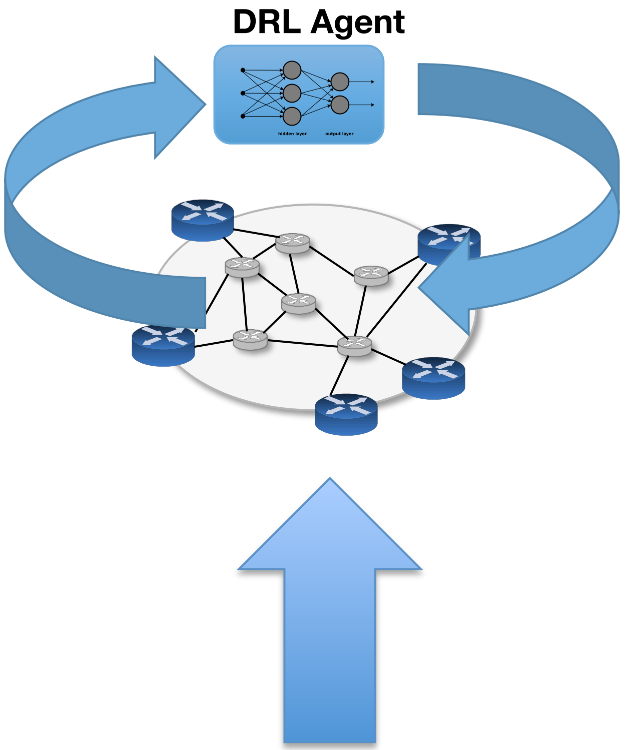
G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntes-Mulero, and A. Cabellos,
"A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization," arXiv preprint arXiv:1709.07080, 2017.

DRL: Operatinal Advantages

- **Fully Autonomous**
  - Does not require prior knowledge of the network
  - Works online and in real-time
  - Learns and optimizes autonomously
- **Advantages** over **traditional optimization** algorithms
  - DRL provides **constant time optimization** vs. the lengthy search process of traditional algorithms
  - **Model-free**: Learns from the environment dynamics, no need for simulation or analytical model.
  - **Black-box optimization**: With DRL agents, different reward functions can target different policies, without the need of designing a new algorithm. Traditional algorithms are taylored to the performance policy.

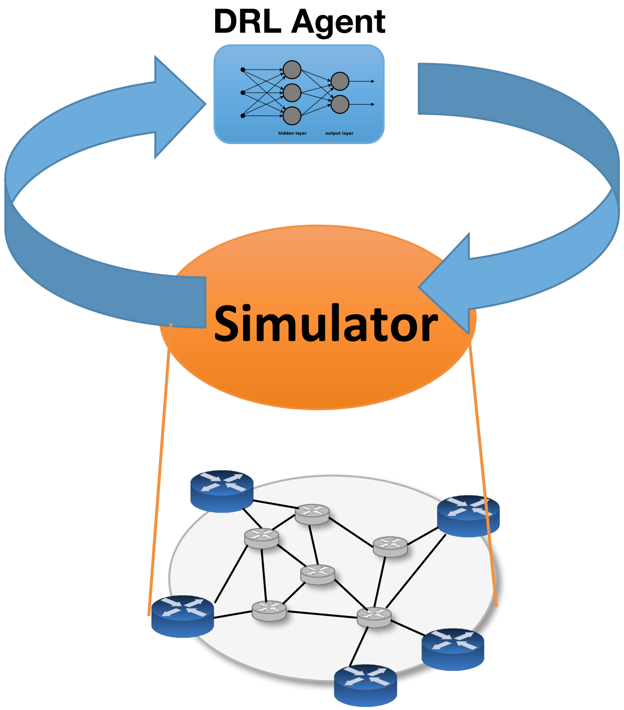
Challenges of DRL: Training

Real Infrastructure

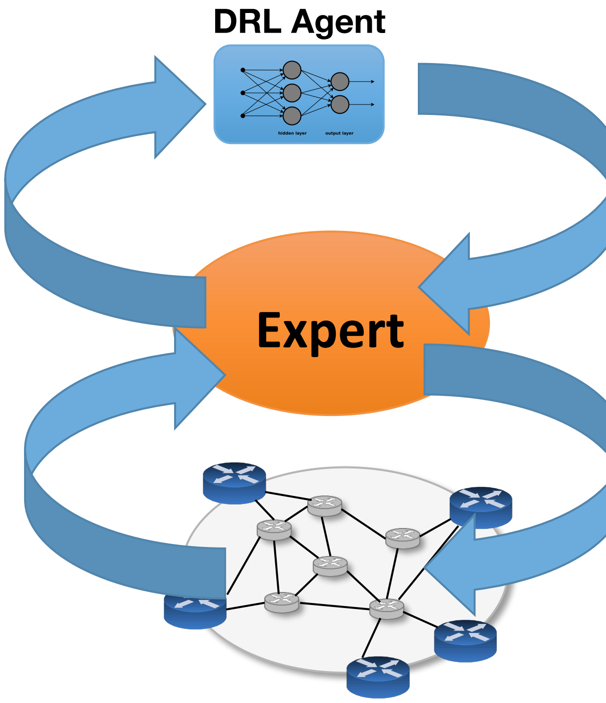


May break your network
During the exploration phase!

Simulator



Expert (Human or Algorithm)



Lack of Explainability

- Deep Neural Networks are inherently black boxes. We don´t know:
  - When will it work, when will it fail
  - Why does it work, why it doesn´t
- No guarantees, no troubleshooting
- Solution: Explainable Artificial Intelligence
  - Aims to develop techniques to develop explainable neural networks

Samek, Wojciech, Thomas Wiegand, and Klaus-Robert Müller. "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models." arXiv preprint arXiv:1708.08296 (2017).

Reward function = Network Management Policy

- The reward function effectively represents, in a mathematical language, the network management policy
- Open questions
  - Can we actually represent any network policy?
    - Are there fundamental limitations?
  - How can we compile existing network policy languages to the mathematical language?

Summary & Conclusions

- Deep Reinforcement Learning represents the full realization of an autonomous intelligent network
- Many advantages
  - Real-time operation (constant-time optimization)
  - Plug & Play (black-box operation)
  - No configuration, just pick the reward function
- Challenges
  - Training: Online, offline or via an expert
  - No guarantees: Towards explainability

Datasets, Code and Papers

- Knowledge-Defined Networking

https://github.com/knowledgedefinednetworking

- DRL for SDN Routing (code and data-sets)

https://github.com/knowledgedefinednetworking/a-deep-rl-approach-for-sdn-routing-optimization

- Work-in-Progress Paper

https://arxiv.org/pdf/1709.07080.pdf