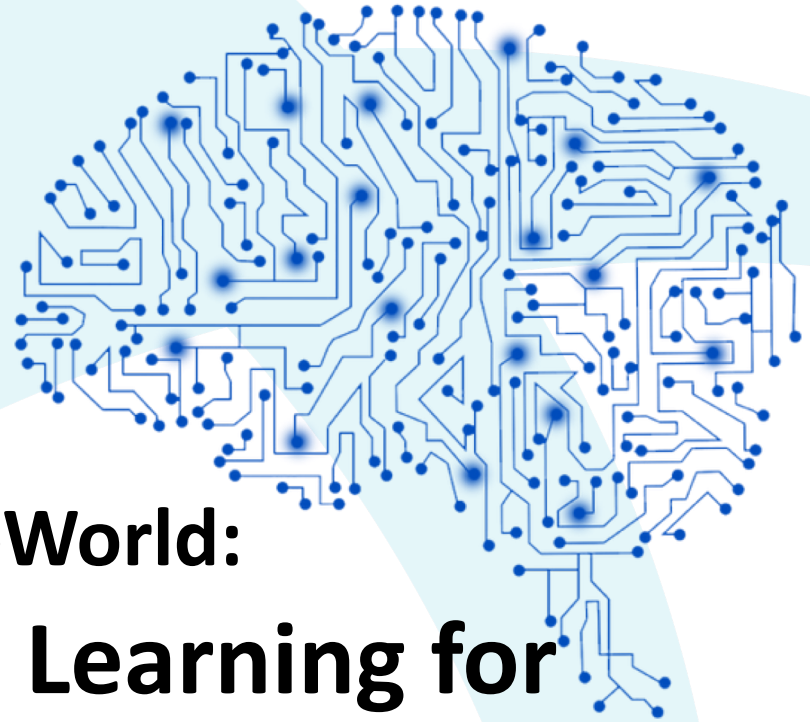
ESnet
ENERGY SCIENCES NETWORK



From Simulations to Real-World:
Deep Reinforcement Learning for Networking

Mariam Kiran

Research Scientist

in Scientific Networking Division

U.S. DEPARTMENT OF ENERGY
Office of Science

BERKELEY LAB

CS Summer Student 2020 Talk

Informal Takeaways

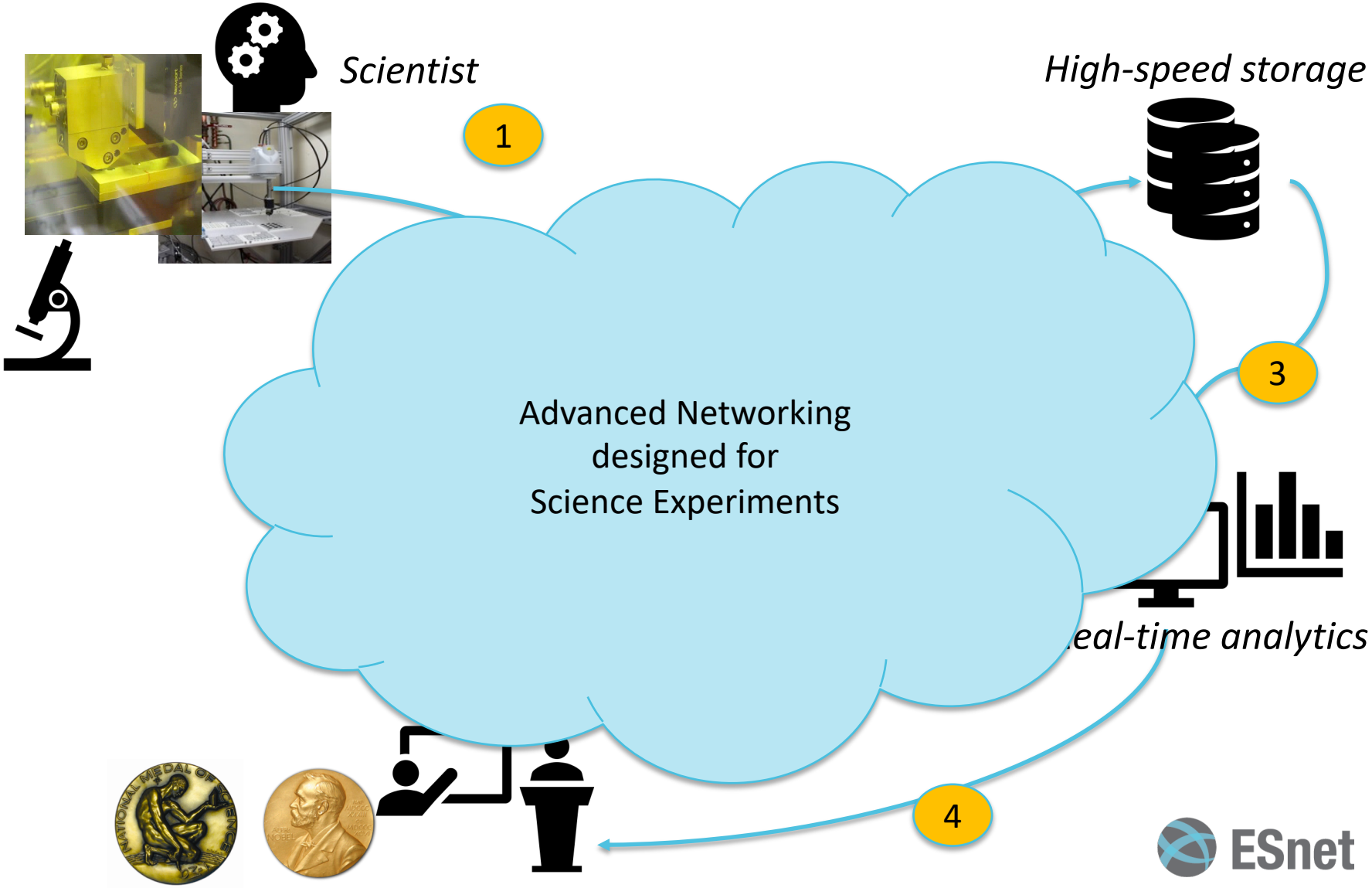
- Networking Research: Is it all solved?
  - Glimpse into how Science R&E networks work
- Can machine (or deep) learning help with some network challenges?
  - Supervised, unsupervised and reinforcement learning*
- What are the challenges?
  - Data issues
  - Experimental testbeds
  - Real-world deployments
- Apart from Networks, what else are we exploring?

ESnet

Networking: A closer look

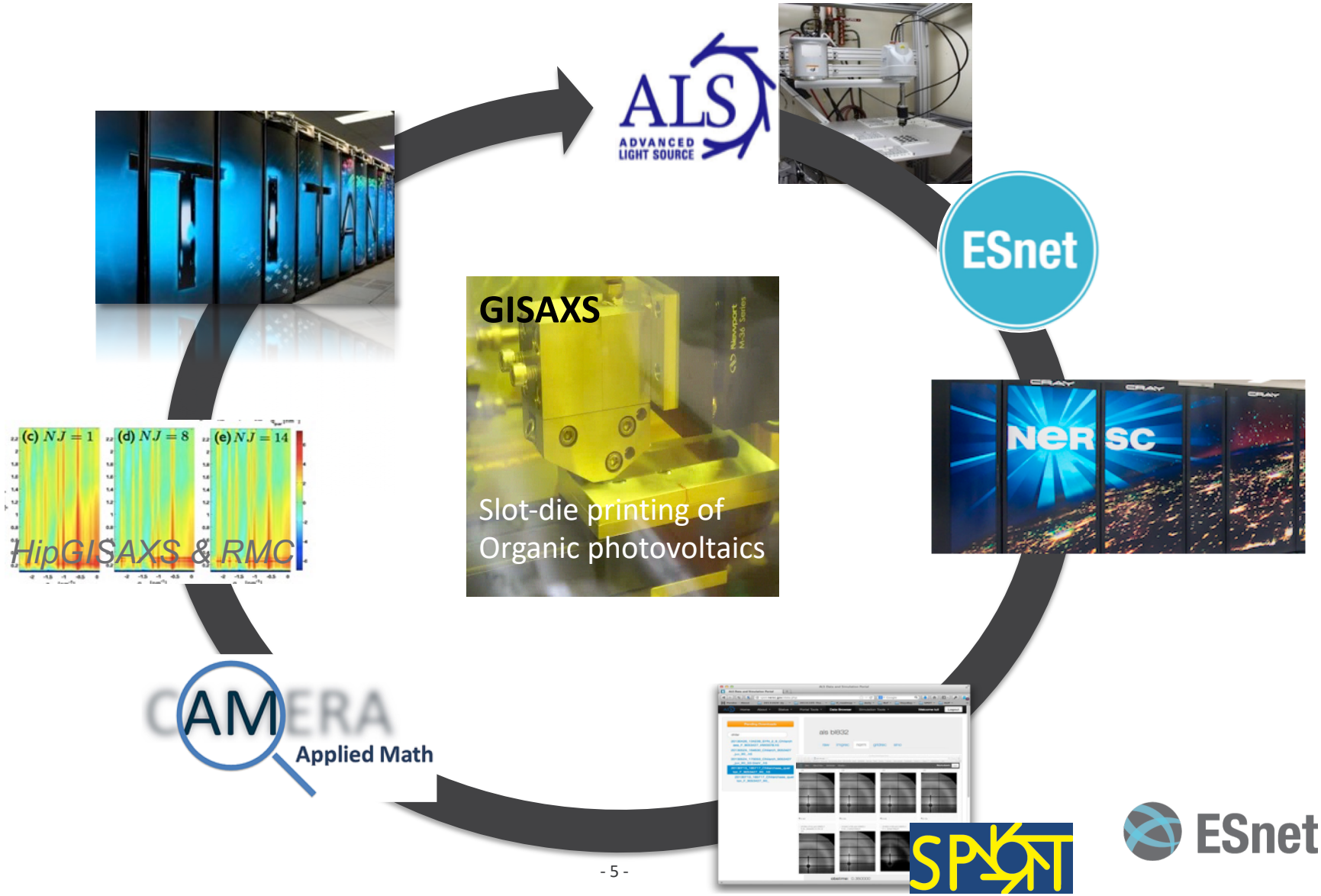
ESnet

Network is the nervous system for Science at Large-scale



Borrowed from E. Dart

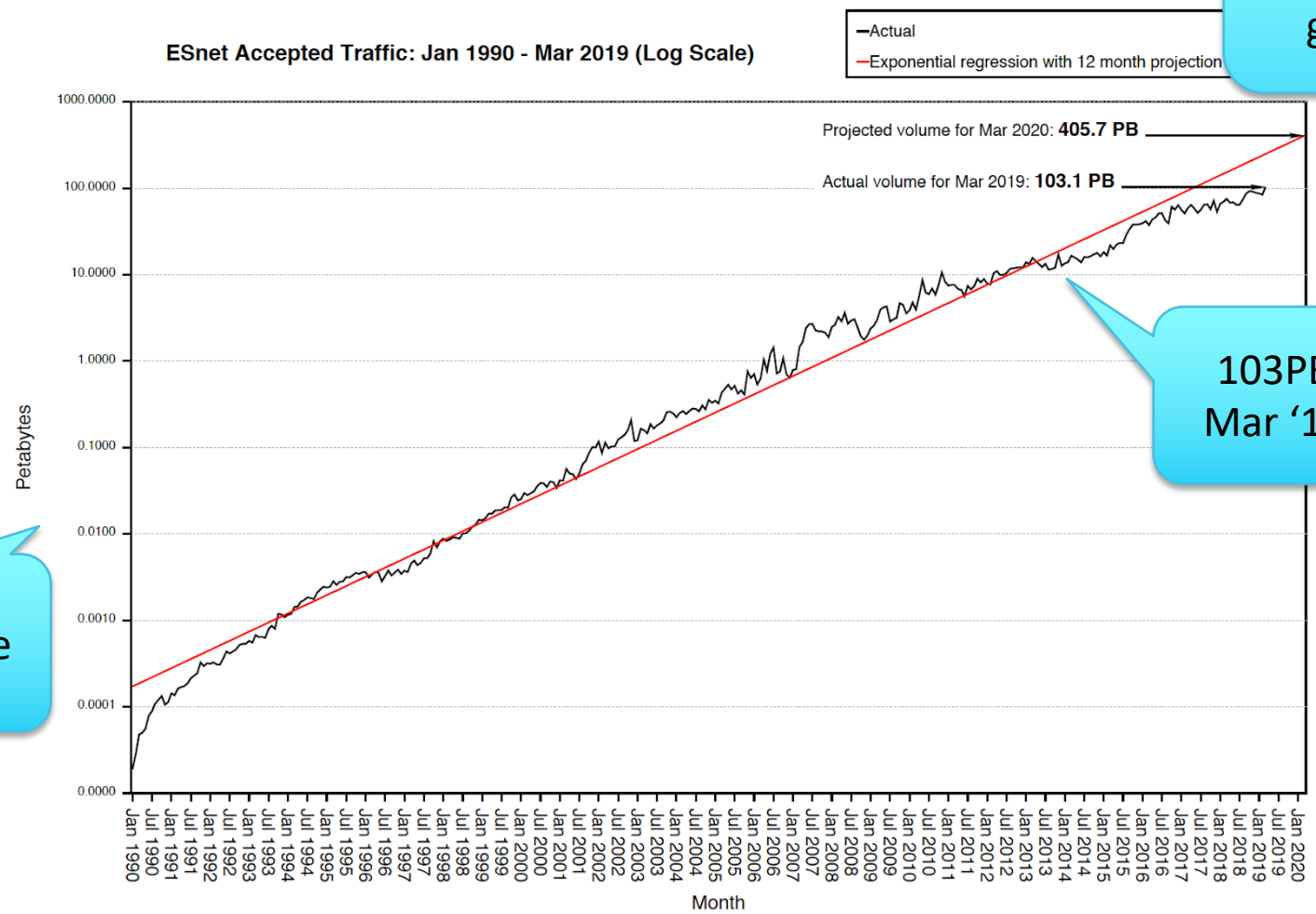
Example Application



Borrowed from C. Guok

Current Science Network Trends

62% yearly traffic growth



103PB Mar '19

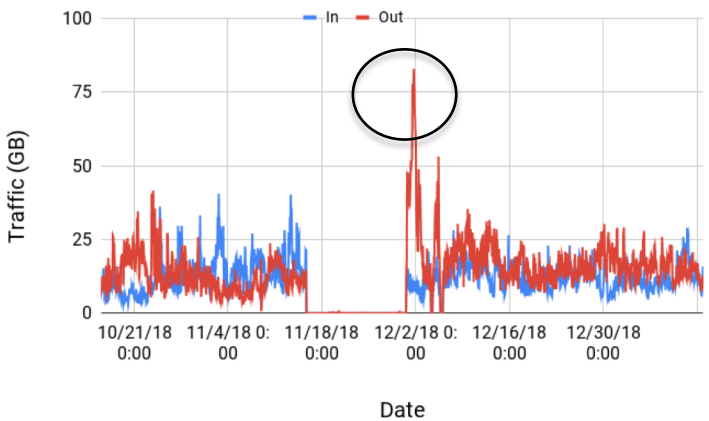
Log scale

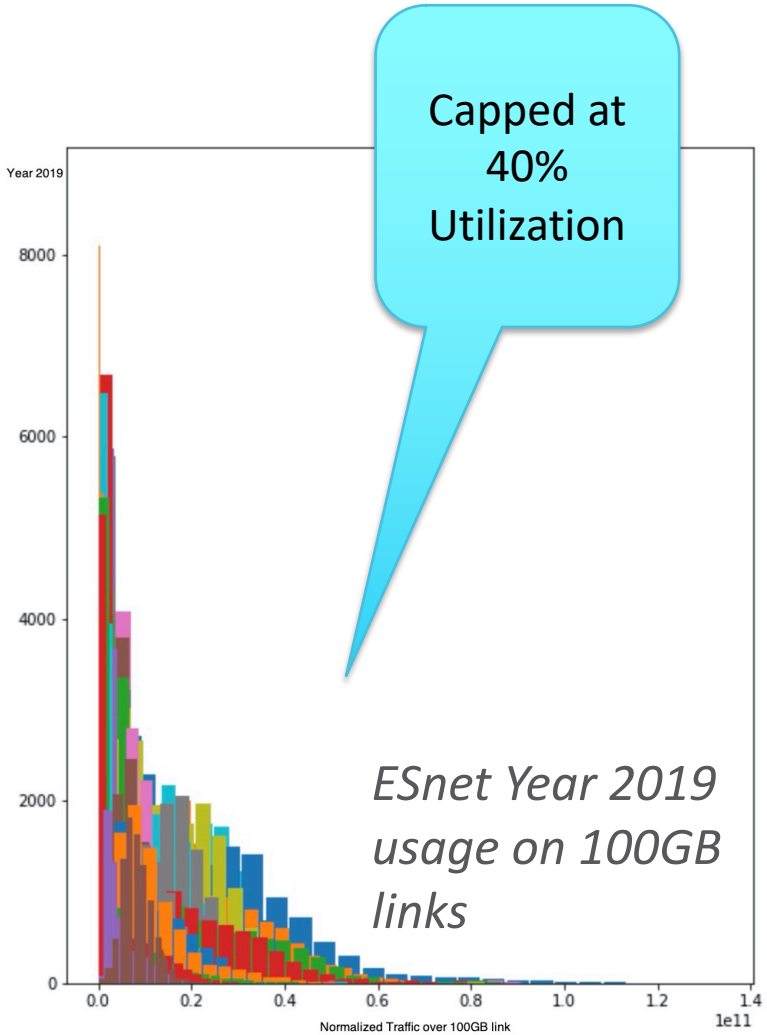
ESnet

Wide area Networks (WAN) *for Science*: What challenges do they see?

ESnet

Resources are often underutilized

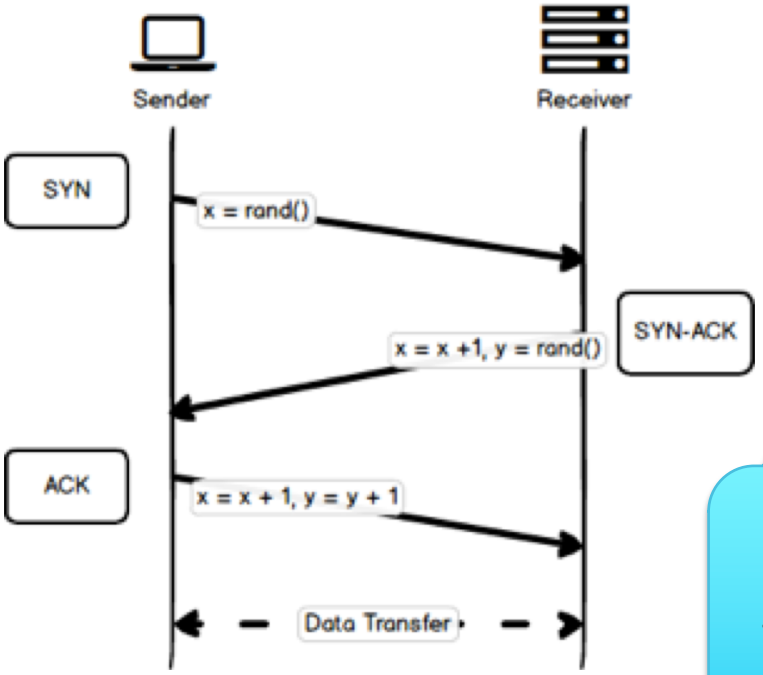


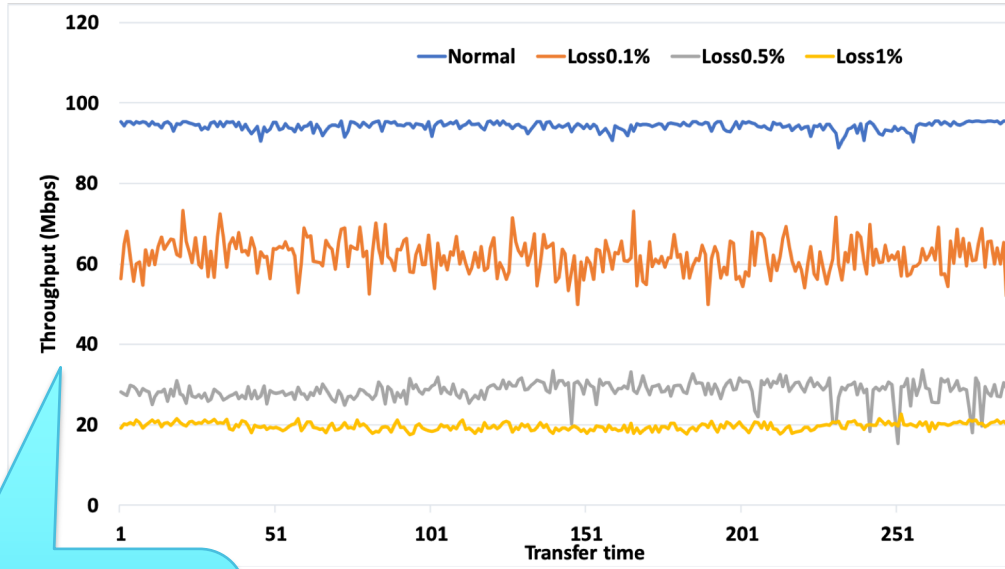


- Networks are designed to be resilient
- Traffic is highly variable and 'bursty' (big versus small transfers)
- Under-utilized resources
- Flow Congestion leads to packet loss

Transfer Performance can be Fragile

- Science uses Transmission Control Protocol (TCP) to send data
- 95% of science traffic uses TCP

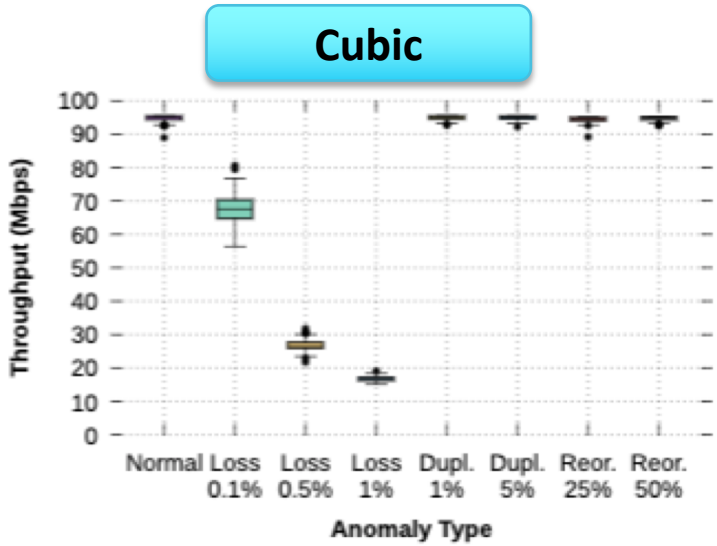


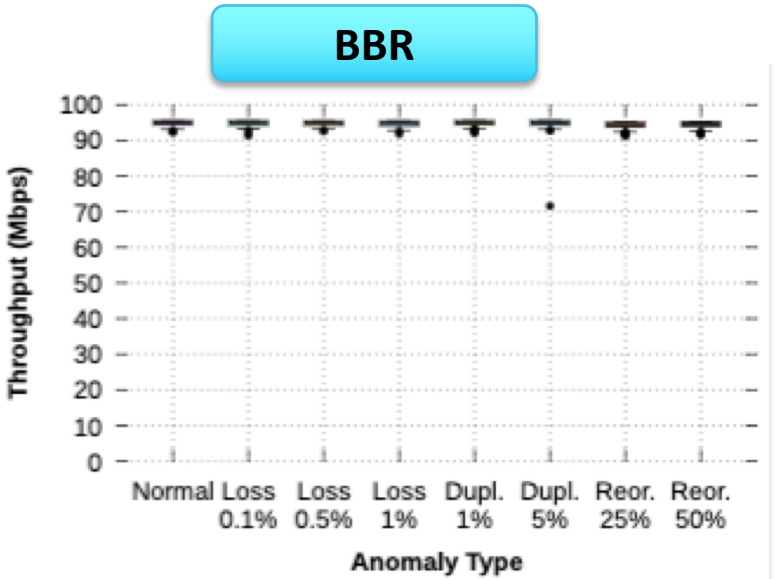


Congestion causes loss that affects throughput

ESnet

Different TCP algorithms for Performance





- Each TCP uses complex algorithms to optimize performance (slow down or quicken data sending)
- Examples include video buffering on Youtube, VOIP (zoom) or big file transfers in few hours (Astronomy/physics from LHC)

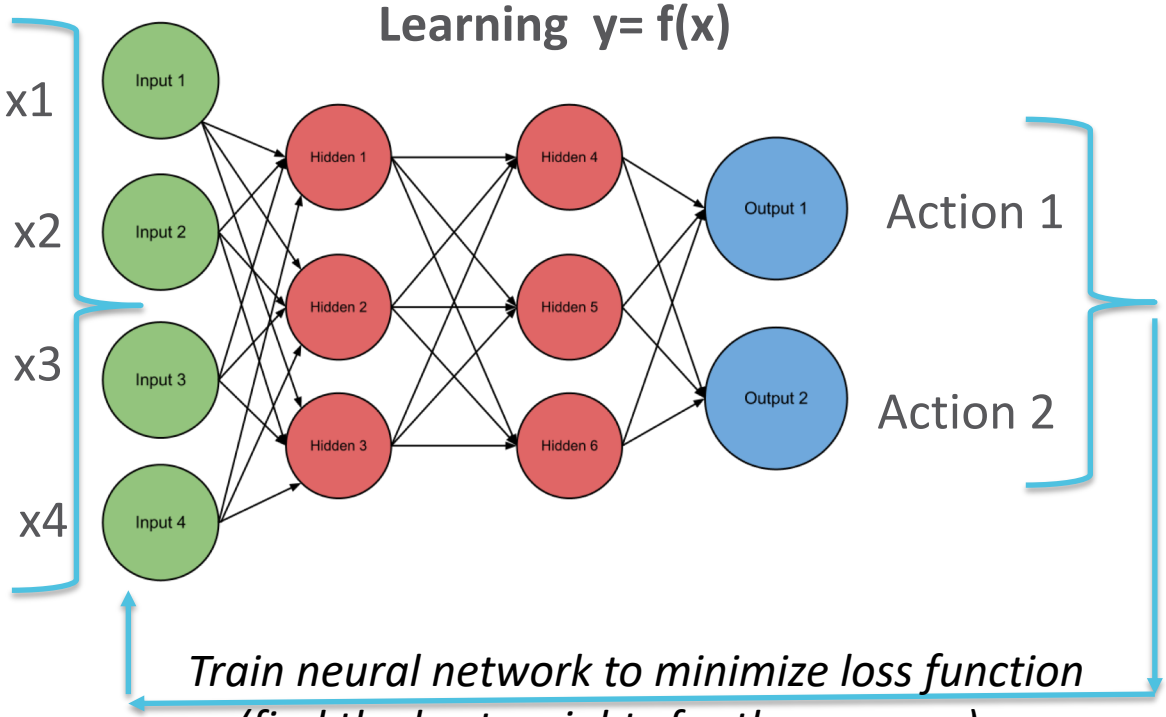
ESnet

ML (Deep) learning to help

Uses of Deep Learning

Objective:

Learning y= f(x)



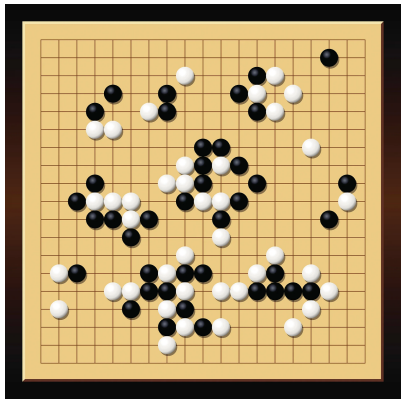
Train neural network to minimize loss function
(find the best weights for the neurons)

Compared to general Machine Learning

Trained to recognize cats

CAT

Trained to play games



Deep learning introduces ‘data-driven learning’ to build bespoke solutions

ESnet

ML and Networking relationship

- Used in many specific applications
  - Security (to identify anomalies)
  - Classify traffic or user behavior on network
  - Recognize problem area

- But a lot of these are on small-scale networks and work in simulated environments
  - Problems with robustness of ML
  - Trusting the ML
  - Understanding how it works
  - And more…

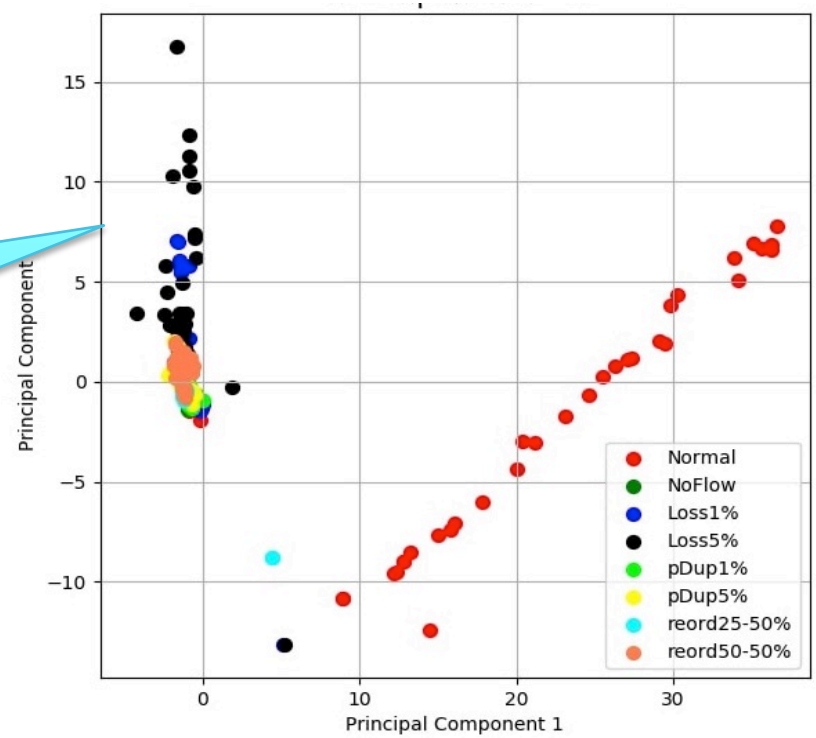
ESnet

Supervised Learning

<u>When Labelled Data is Available</u>

- Learning to recognize the labels
- Classification, object detection, anomaly detection
- Works very well if we can identify clear class boundaries

Good versus bad flows



Plot of anomalous TCP traffic

ESnet

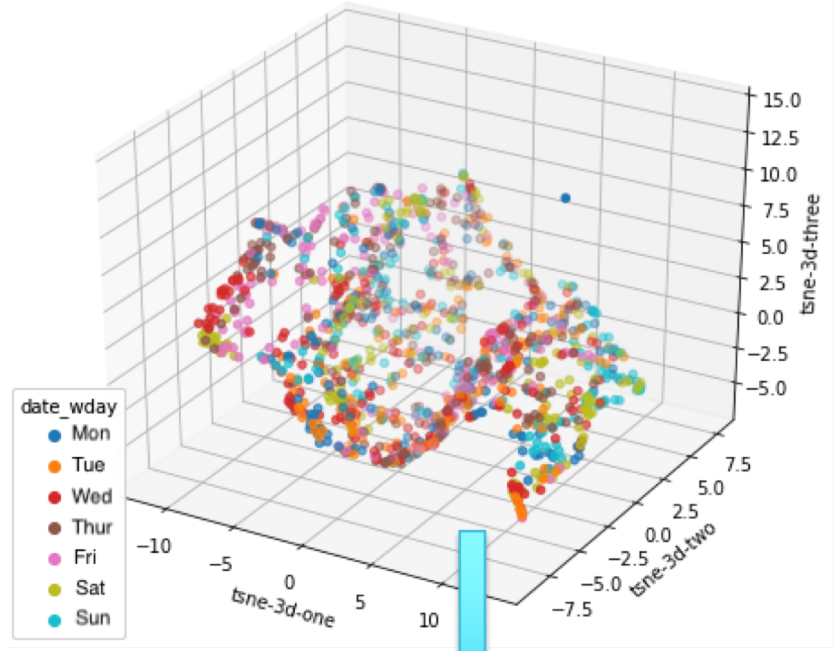
*S. Campbell, F. Bannatwala

Unsupervised Learning

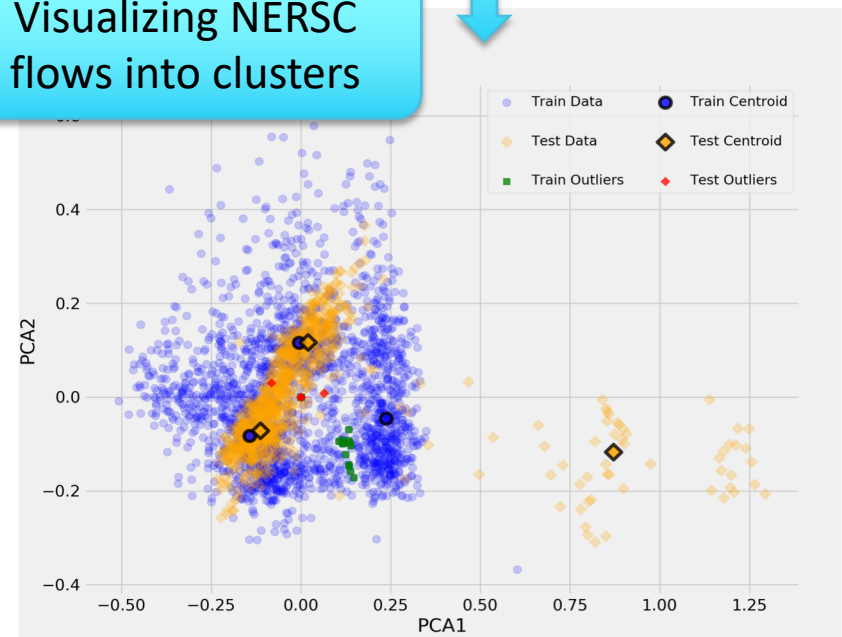
When NO labelled data in Available

- Learn underlying rules in the data
- Clustering, feature identification, recognize anomalies in test data

- Works well when working with domain scientists after clusters are recognized



Visualizing NERSC flows into clusters



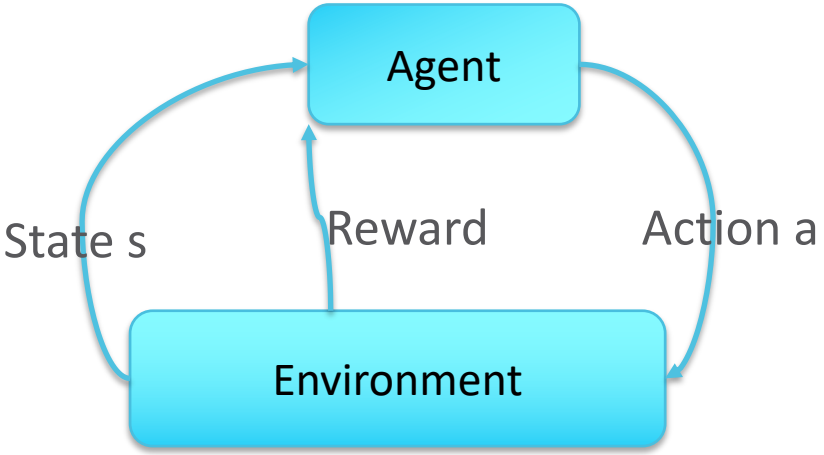
Plot of NERSC TCP flow transfers

Reinforcement learning (RL)*

<u>When NO data is available</u>

- Learn via trial and error
- Interactions with the environment

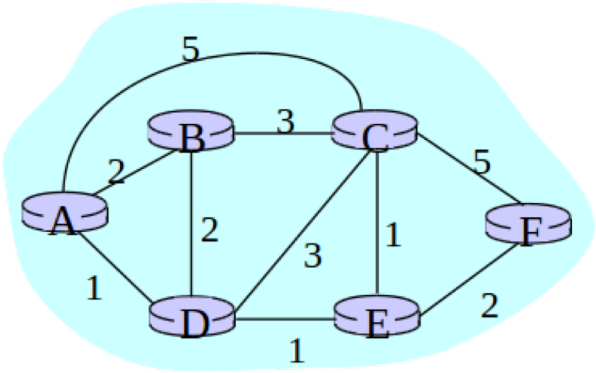
- Agent views the state of the environment and chooses an action
- Each action, agent gets a reward
- Over time, agent learns optimal actions that give best rewards and what to do next



ESnet

Mapping Deep RL to Networks

RL for Traffic Engineering



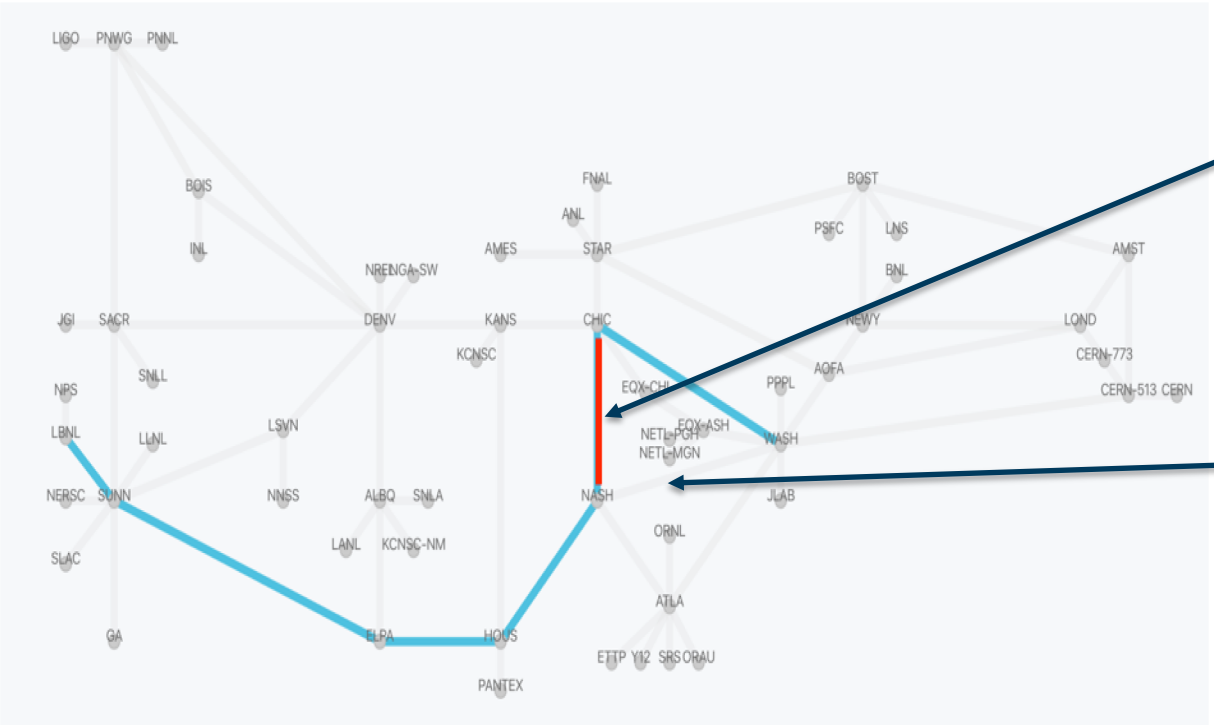
- Link-state (Dijkstra)-OSPF
  - Shortest possible route
  - Complete view of network topology
  - Central traffic engineering
- Distance-vector (Bellman-Ford)-RIP
  - Routers only know their neighbors
  - Shortest possible route to destination via iterative calculations
  - Each node updates routing tables

- New opportunities with Software defined networking + RL
  - RL decisions to determining best routes

ESnet

Network ‘Self’ learns to reduce Congestion

Too many flows on same path → Congestion → Packet loss



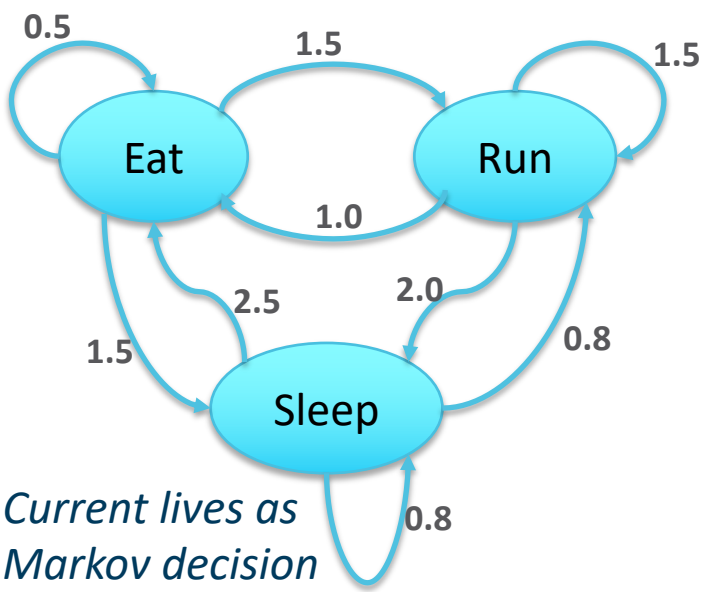
Predict if congestion will occur

Divert traffic to underused paths

ESnet

But first …… Working with Dynamic Systems

- Supervised and Unsupervised learning:
  - Good to find patterns in data sets
- Reinforcement: Teach systems how to perform (i.e. games, controlling systems, dynamic systems)



Current lives as Markov decision process

Possible sequence of events:

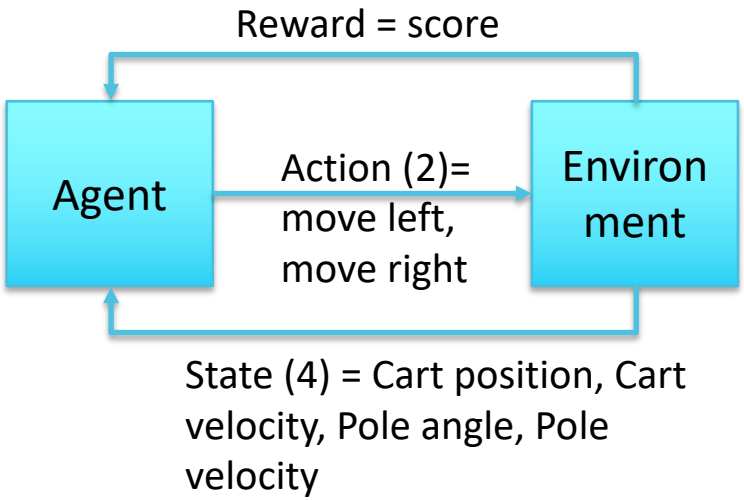
- Eat-Run-Sleep
- Run-Sleep-Eat

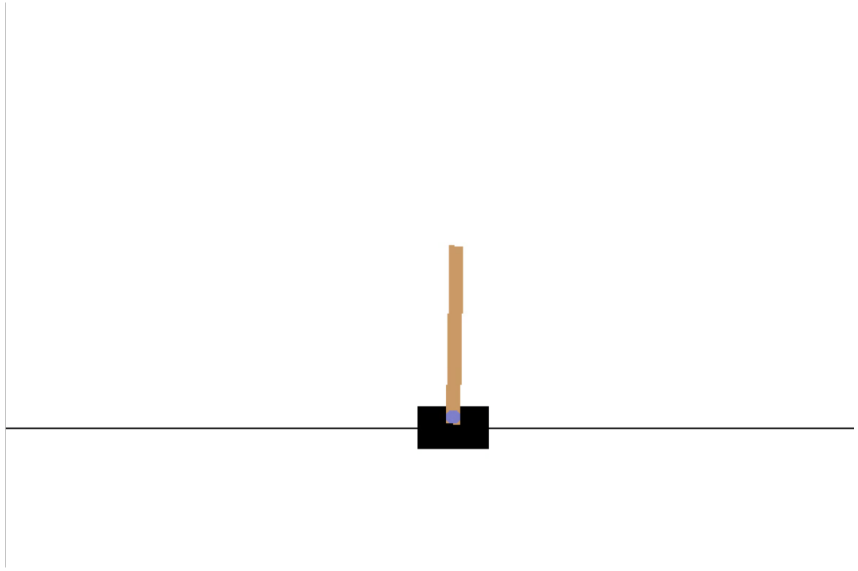
Total Reward = $r_{t=1} + r_{t=2} + r_{t=3} \ldots + \ldots + r_{t=T}$

Train yourself to get the max reward

Useful in Games: Cartpole Example

- Inverted pendulum problem (control theory)
- Goal: balance the pole upright





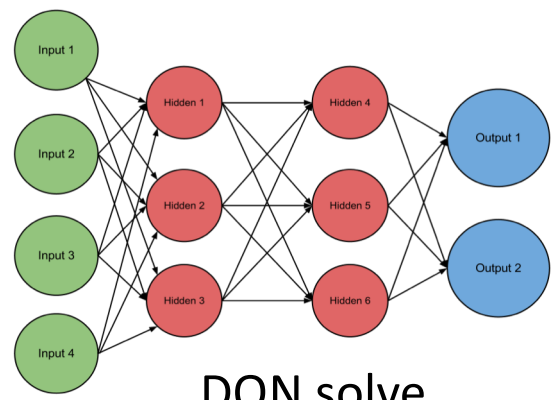
Cartpole (2): Learning state-action pairs

- Learning the Q function (value function or Bellman equation):

$$Q(s,a) = R(s,a) + \gamma \max_{a'} Q(s',a'))$$

- When state and action space explodes, replace with a neural network

state	action	value
state0	Move left	10
state1	Move right	2
…	…	..
…	..	..



DQN solve
(state, action, reward, state_next)

Episodes and Experience

- At time t,
  - Agent sees environment state, $s_t \in S$
  - Agent chooses an action to perform, $a \in A(s_t)$.
  - As a consequence of its action, the environment changes its state, $s_{(t+1)}$
  - Agent receives a reward or payoff, r_t.
- Episodic learning over time

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha(R(s,a) + \gamma \max Q(s',a') - Q_{t-1}(s,a))$$

- Agents keep 'experience-replay' buffer to remember past actions
- Hyperparameter optimization*:
  - Alpha: Learning rate, how quickly can agent learn
  - Gamma: prioritize immediate rewards versus future rewards
  - Epsilon: exploration factor

** Upcoming paper on effect of hyperparameters on RL*

ESnet

Different RL algorithms

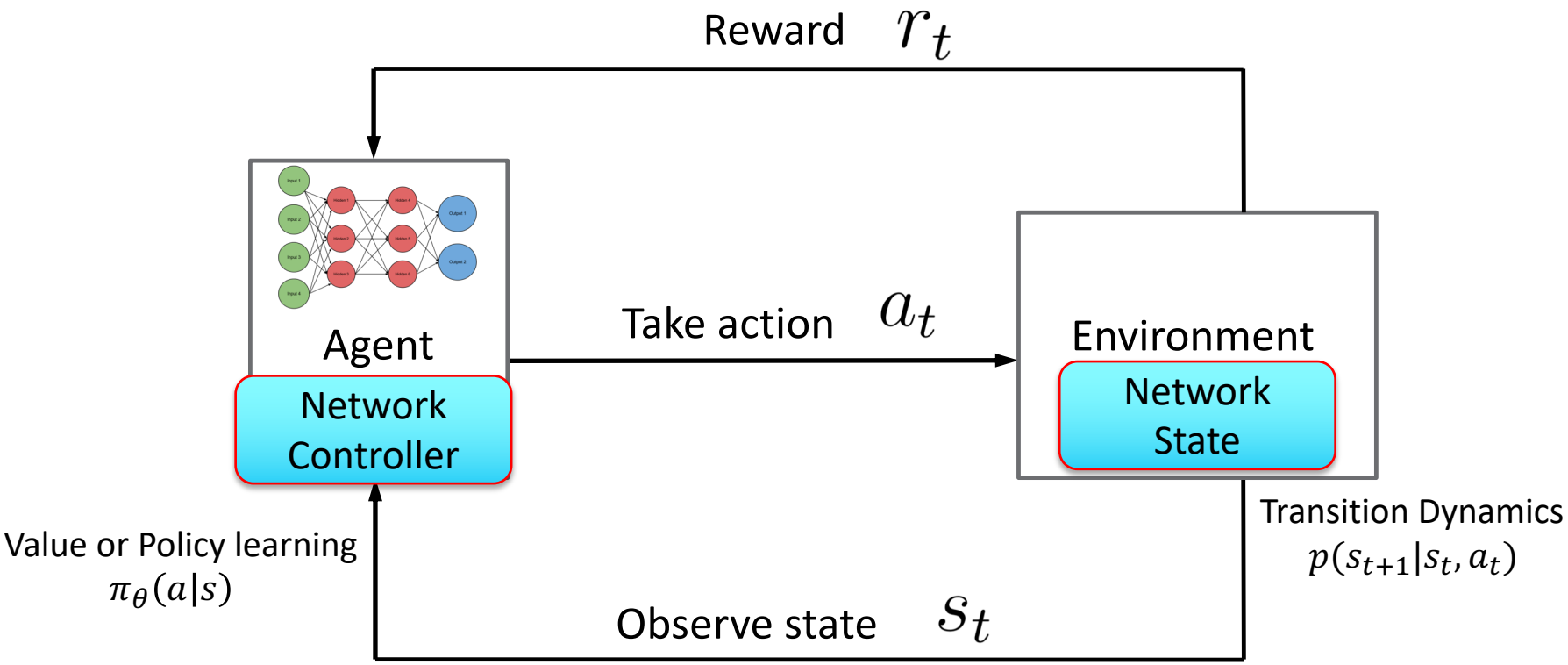
Technique*	model	policy	Action space	observations	operators
Q-learning	free	Off-policy	discrete	discrete	Q-value
DQN	free	Off-policy	discrete	continuous	Q-value
DDPG	free	Off-policy	continuous	continuous	Q-value
TRPO	free	Off-policy	continuous	continuous	advantage
A3C	free	Off-policy	continuous	continuous	advantage

- Most are developed in robotics and games (All work in simulated world)
- No real-world deployments

ESnet

Challenge:
Conversion to a Learning Problem

Neural Network Learning a (WAN) Network

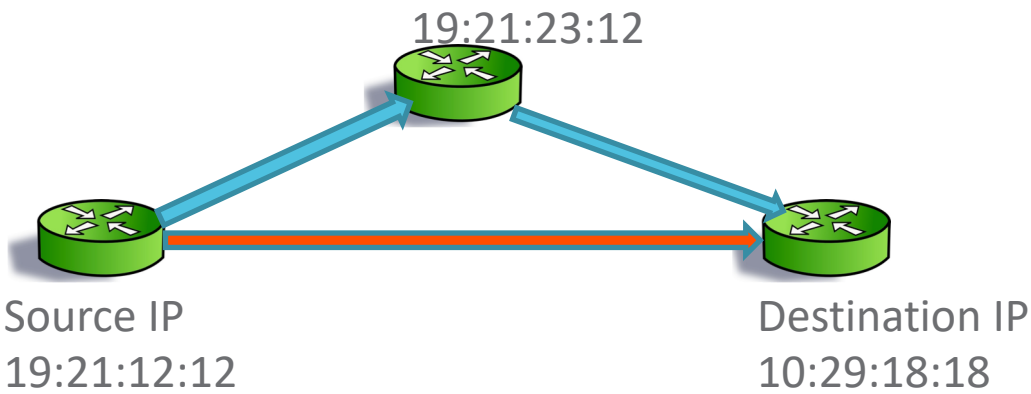


Goal: maximize the cumulative reward $\sum_t r_t$

Each time step, RL agent collects states, generates an action and updates policy based on reward

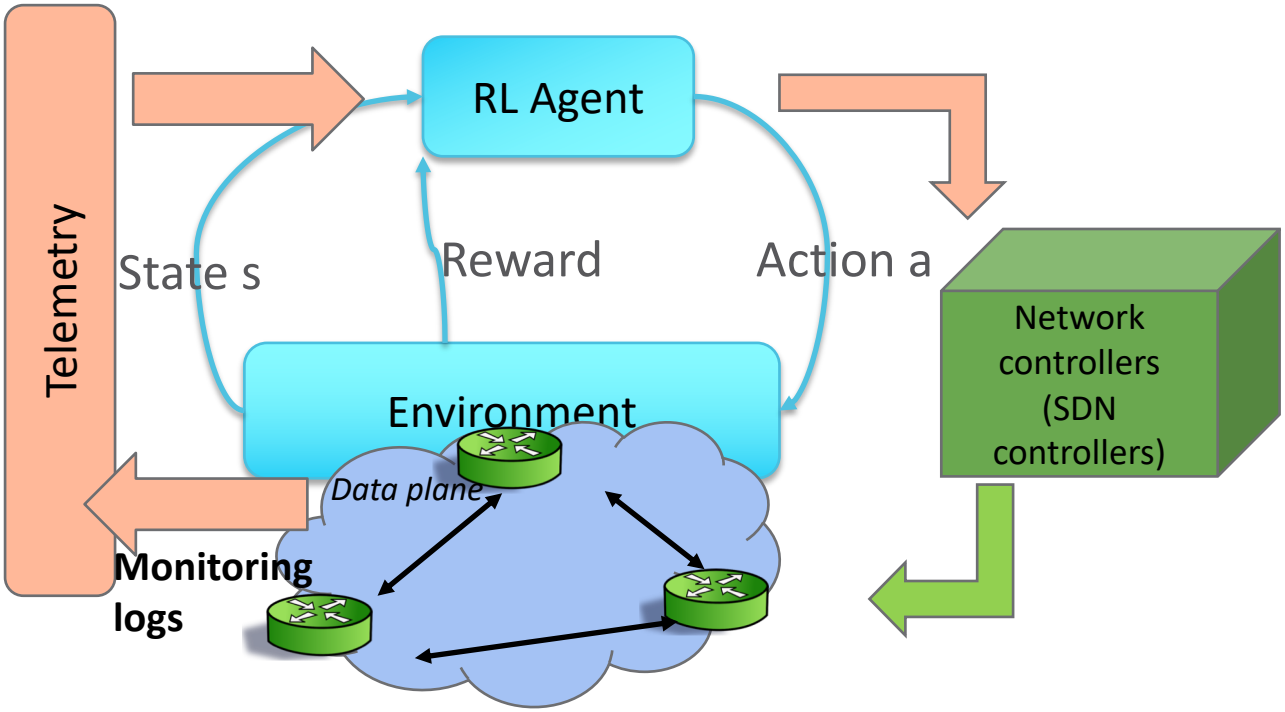
Convert to Learning problem

- Networks have multiple paths between source and destinations
- Convert the problem into Markov decision process:
  - (S, A, R, P), where s ∈ S set of states, a ∈ A set of actions, R(s, a, s') represents reward and probability P for executing action in state s.

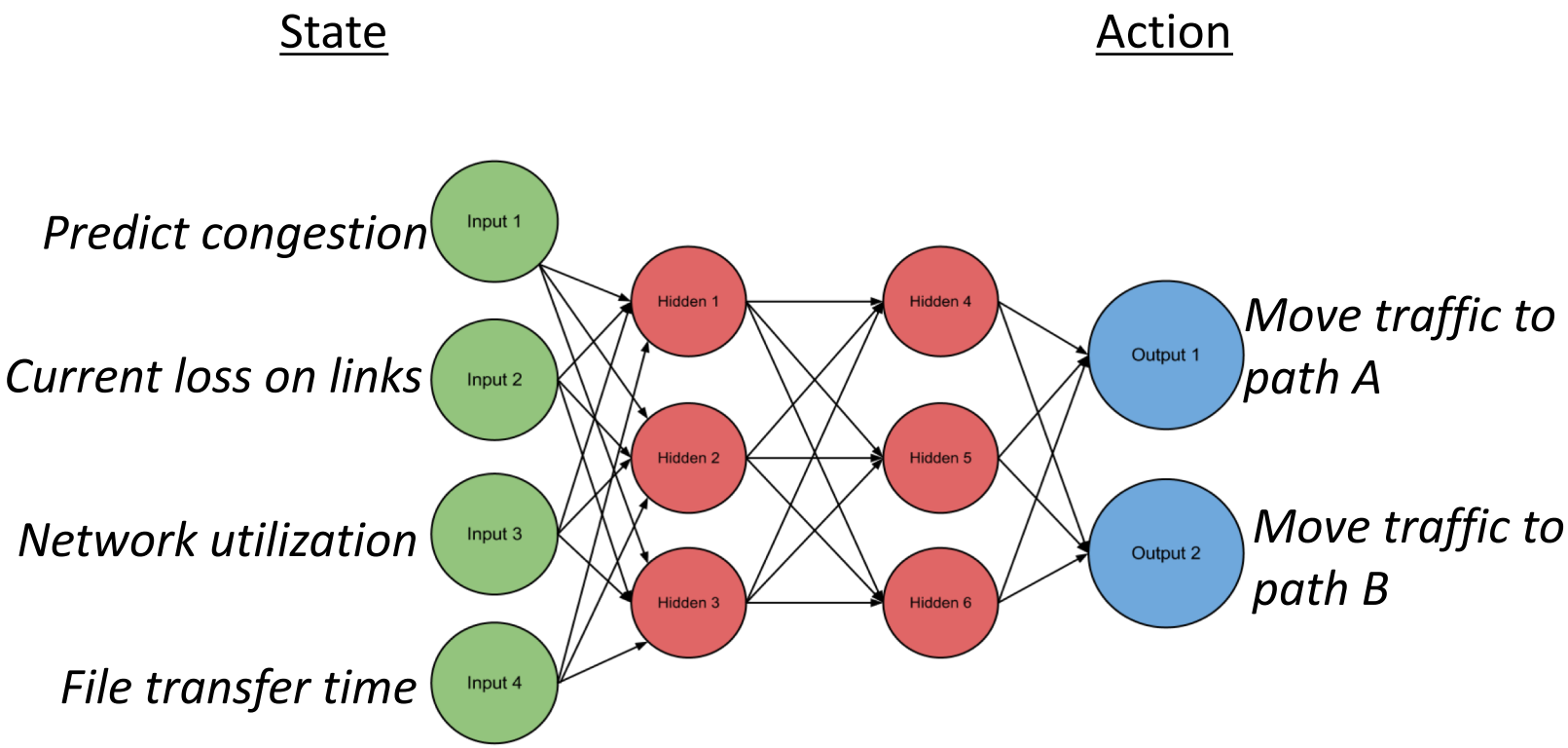


- RL agent at source: Chooses best path to send flows
  - Trial-and-error to learn "optimal" paths: not just shortest path
  - Once learned, it knows which one to choose for arriving flow

Links with Software Defined Networks (SDN)



Designing the Neural Network



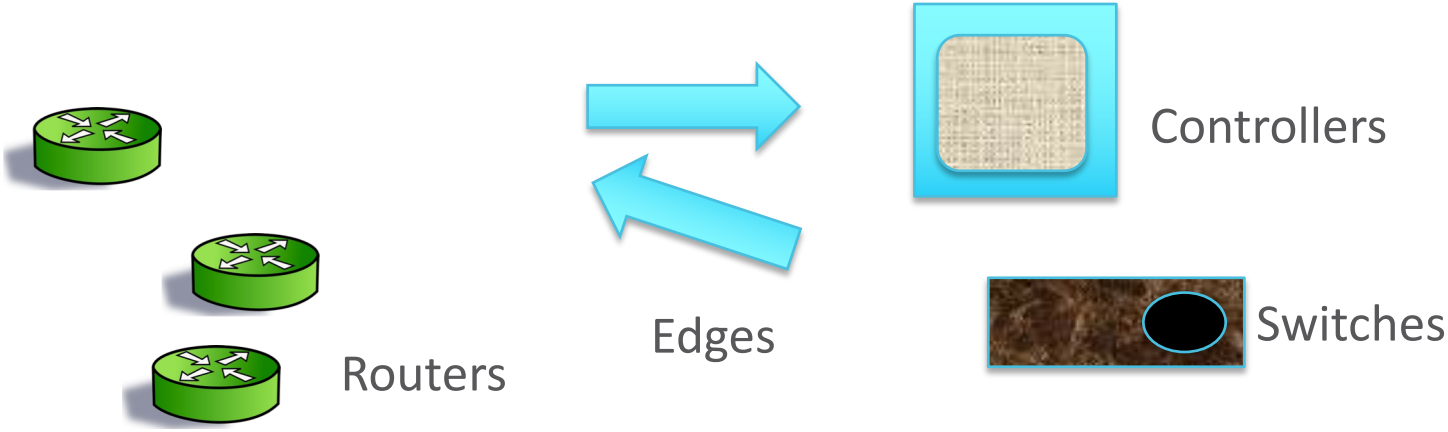
Challenge:
Designing the Training Environment

Working with Network Emulators

- Mininet, NS2, CloudSim, examples of simulating network behavior
- Real-world is very different from simulations/emulator
  - Noise,
  - lack of data,
  - too diverse data tools,
  - connection challenges
- Emulators not designed to test ML advances
- Do not scale to our problem

ESnet

Designing our own Network Gym: NetGame

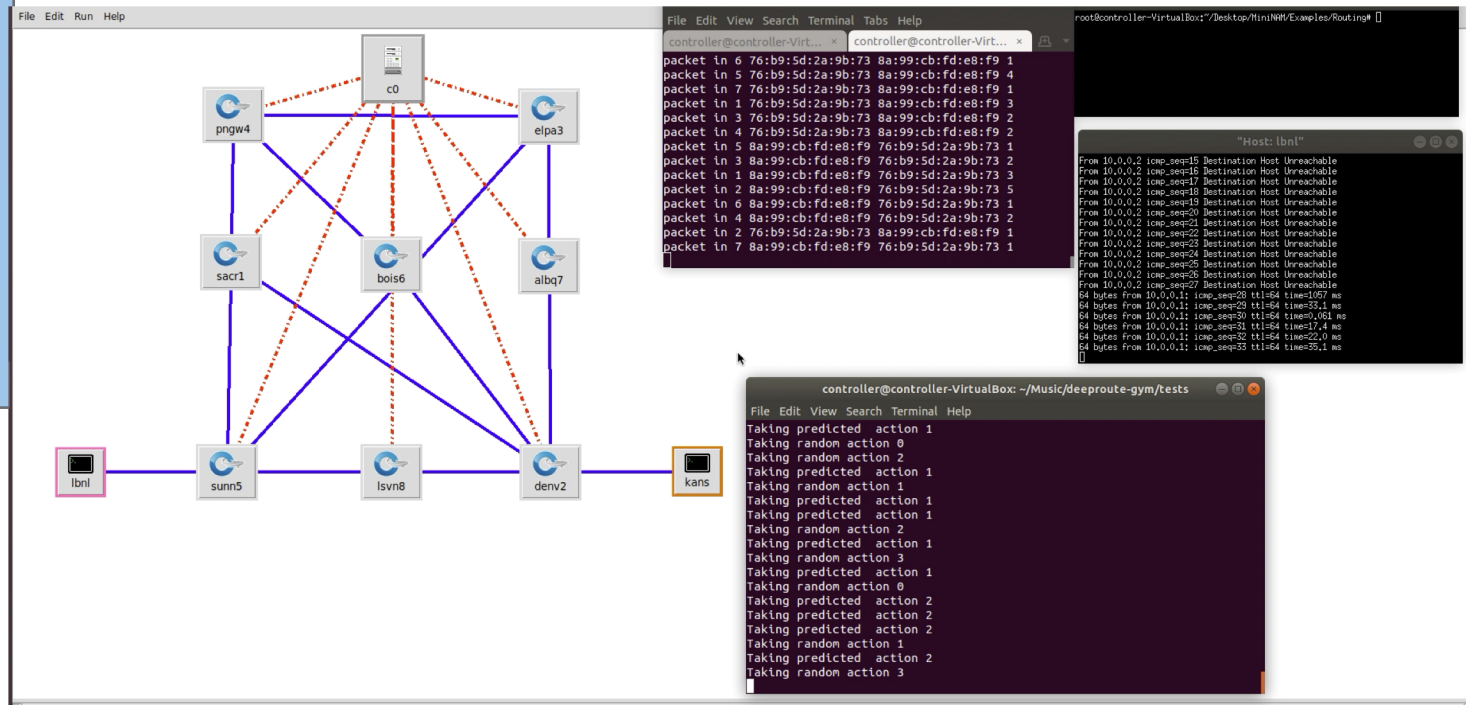


- Agent-based modelling
- Simulating traffic with complex needs
- Scaling network up dynamically (complex scenarios, Multi-RL)
- Link in real data sets to fuel our simulation
- More control to understand why ML decisions are being made

ESnet

Deploying RL in Mininet Simulation

State: Current network utilization
Action: Path to take
Reward: Flow completion time
Optimize: best path



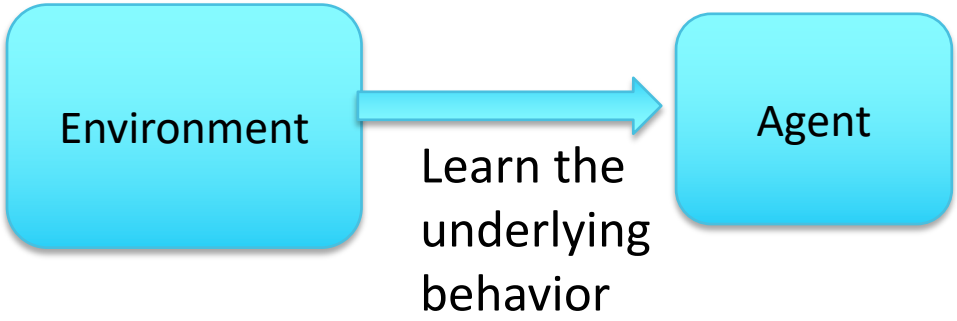
*DeepRoute: Herding Elephant and Mice Flows with Reinforcement Learning, MLN'2019

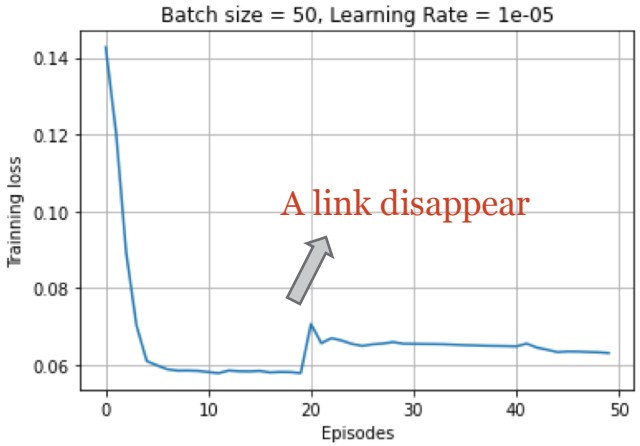
ESnet

Challenge:
Working with Dynamic Data

*S Sun, Summer Intern

Networks Change

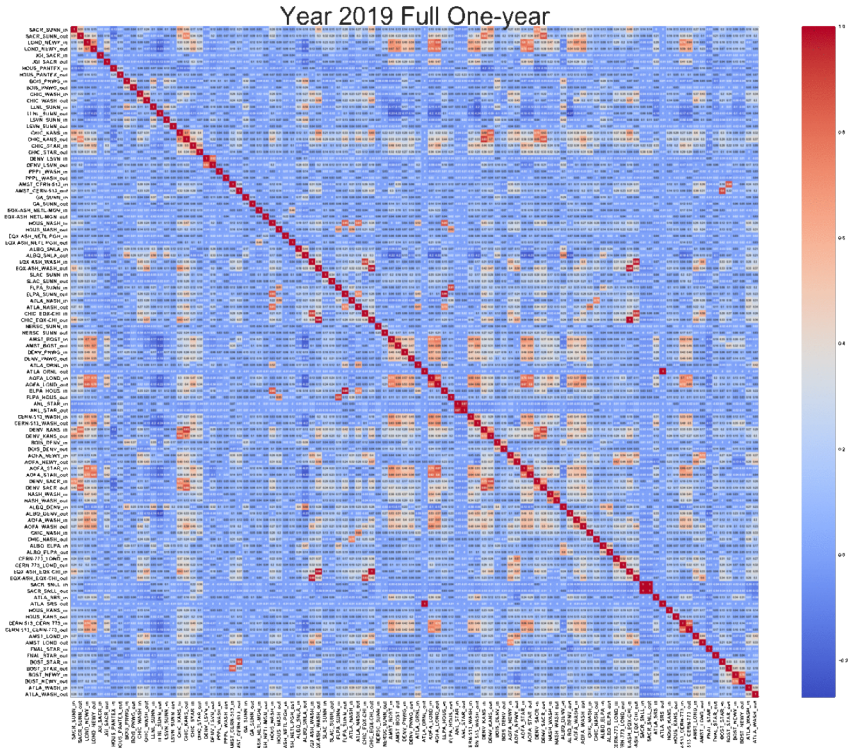




- Can the DRL Agent adapt?
- Model Predictive Control to learn underlying physical laws in the system

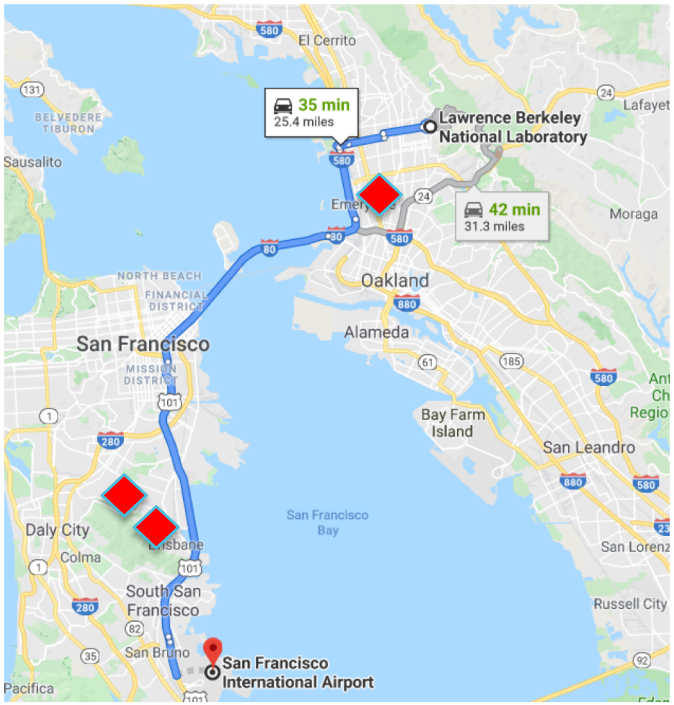
Data > underlying laws > Agent

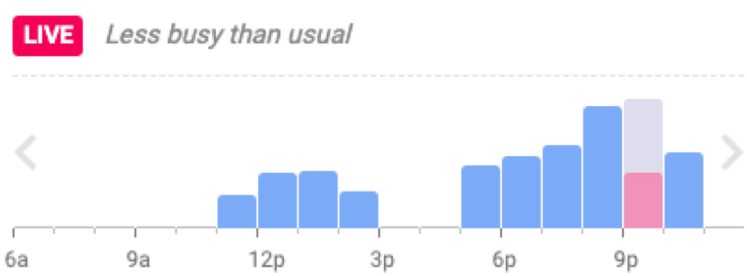
Traffic patterns Constantly Change

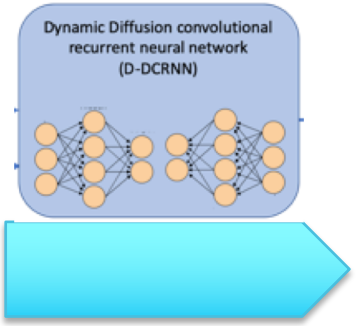


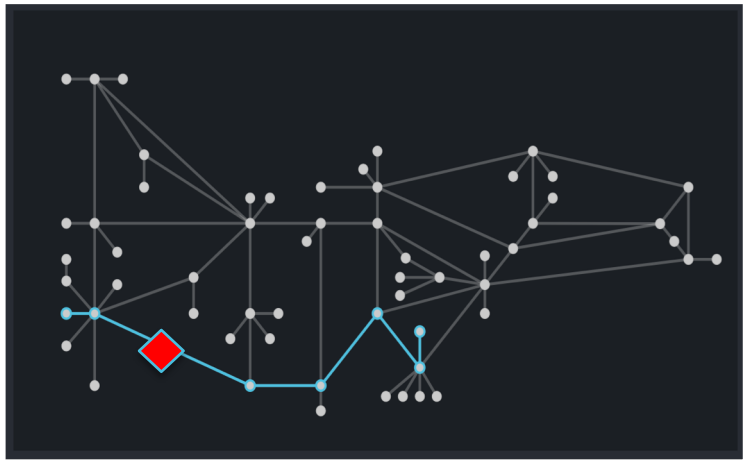
- Correlation of traffic over year 2019
- No underlying patterns, seasonality because science transfers are erratic
- LSTM models were better to learn short term predictions. Long term patterns don't affect us.
- Feed into graph neural networks to enhance predictions

Example: Predicting Congestion Ahead of Time









With Real-time Data

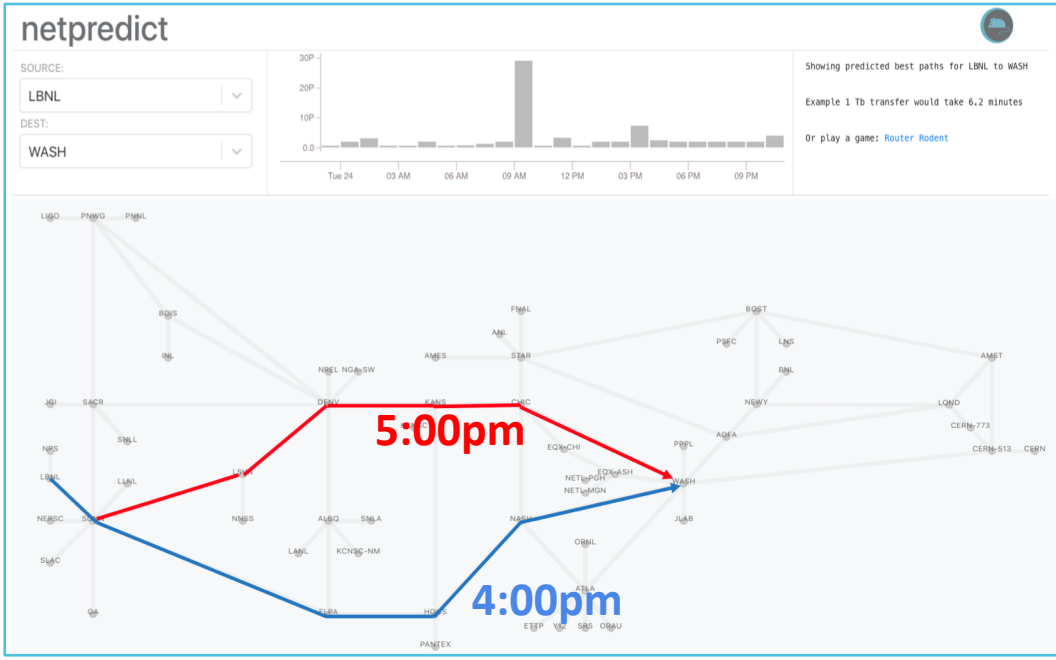
- Loss - Perfsonar
- Traffic statistics - SNMP
- Logs – NetFlow

ESnet

NetPredict for Just-in-Time Flows
(Linked with ESnet portal)

*SC19 Demo Network Research Exhibitions

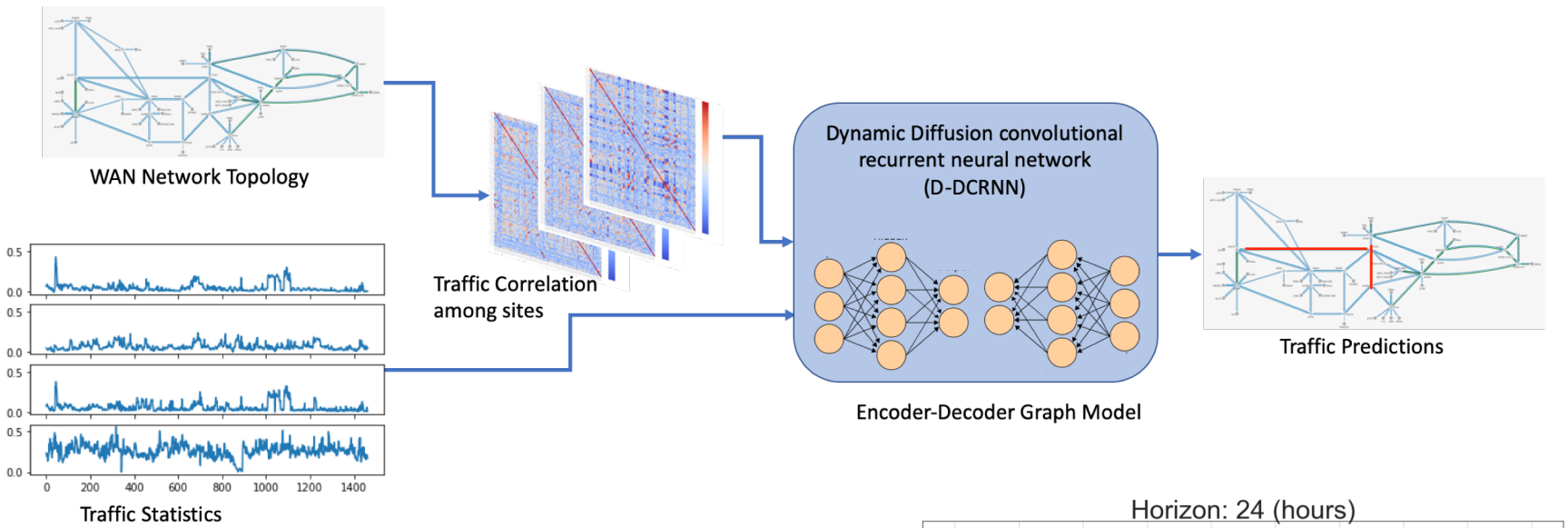
- Deployed on Google Cloud Platform
  - Different models can run at the same time to compute least congested paths
  - Estimates transfer completion time
- Trust dashboard
  - Real-time ML performance
  - Build engineer's confidence in predictions



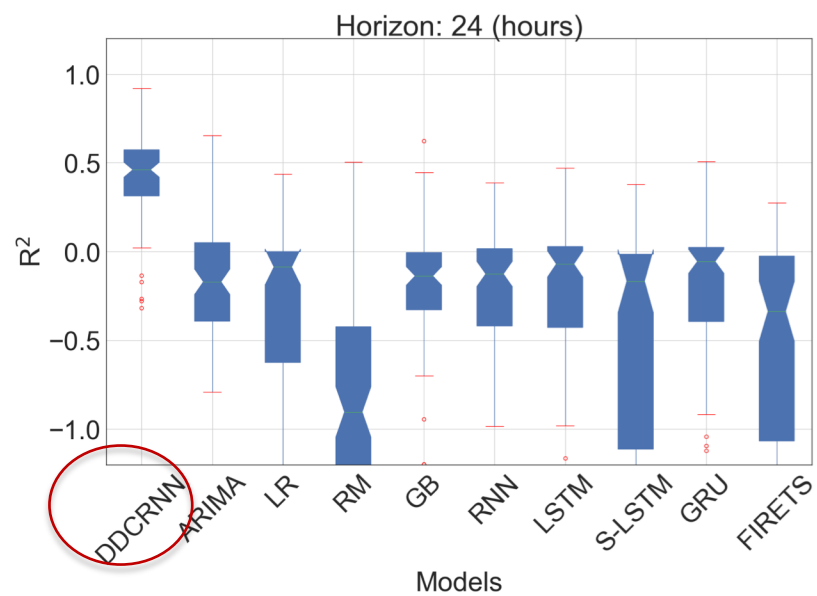
ESnet

* P. Balaprakash (ANL)

Graph neural networks to improve predictons



GNN high order of accuracy compared to simple LSTM approaches

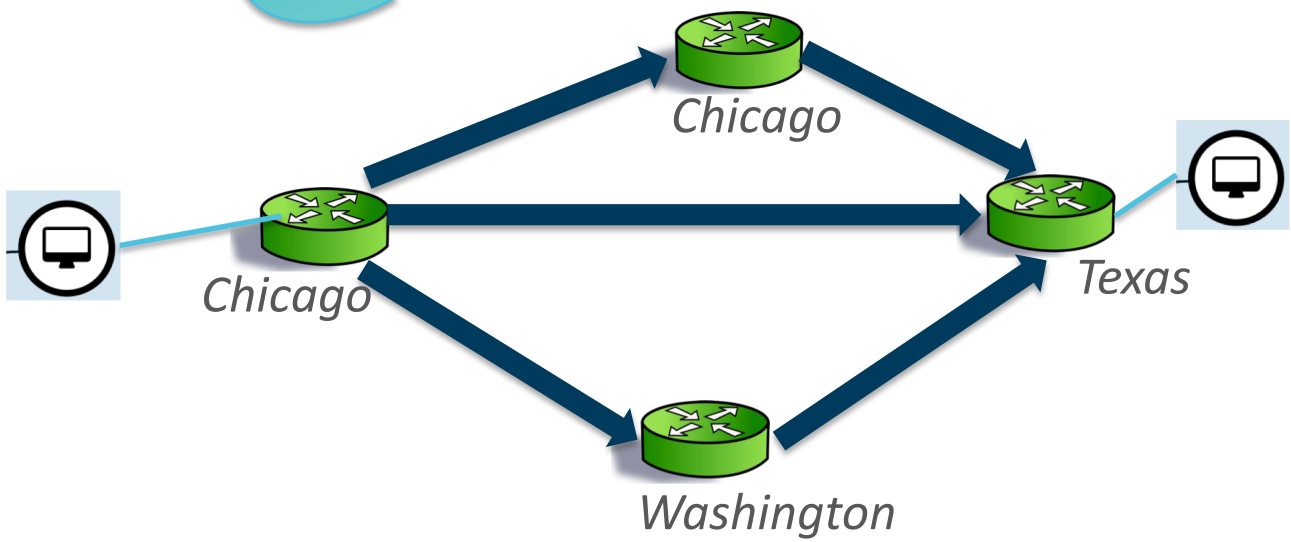


*Data-driven Learning to Predict WAN Network Traffic, SNTA'20

Challenge:
Deploying Theory to Practical

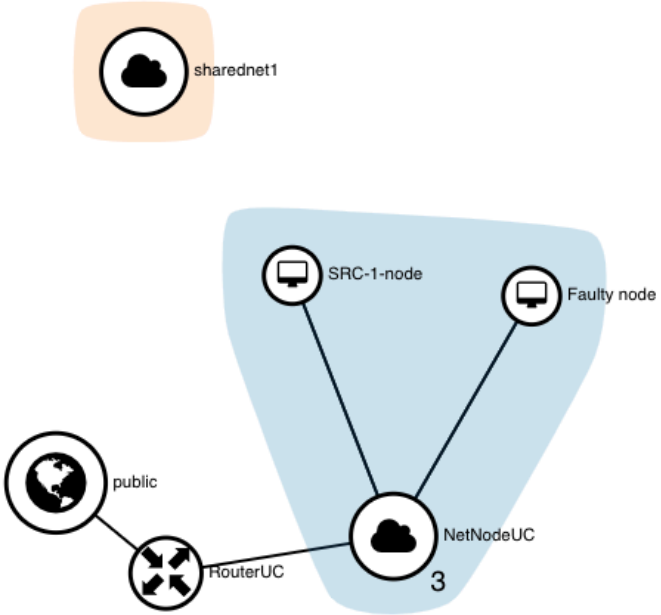
Building Our Own Topology

A lot of our time is consumed by building real networks to test our algorithms



- ESnet and other network data
- lack of labelled data

Gathering data, OpenStack nodes …



Displaying 5 items

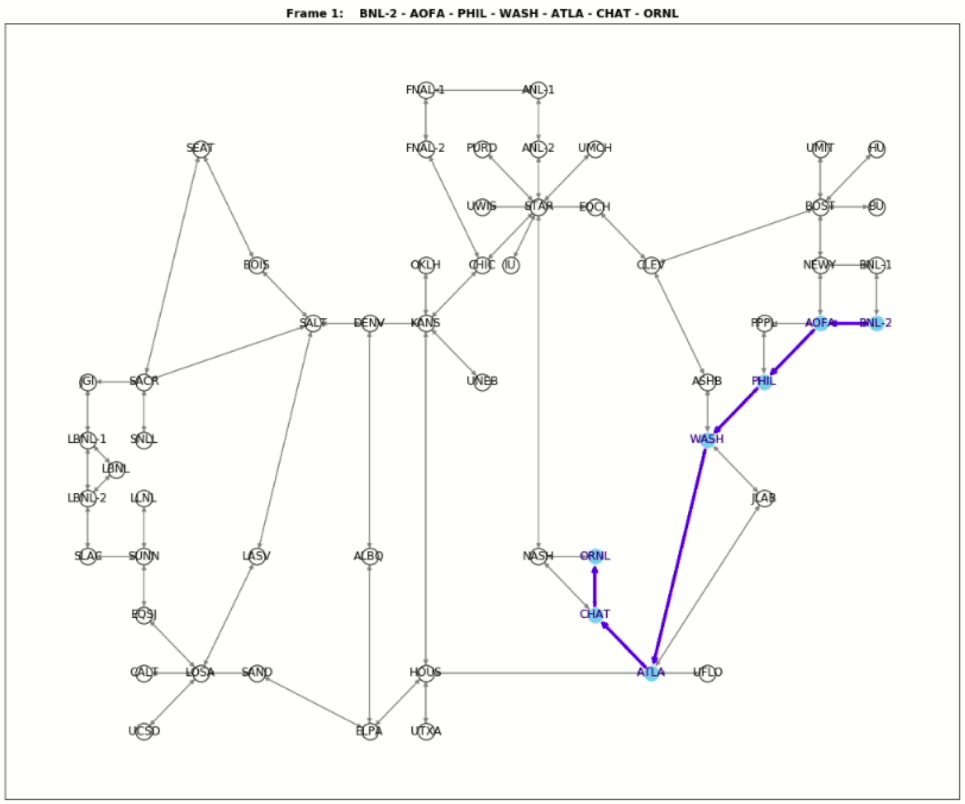
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
perfSONAR-node	CC-CentOS7	192.168.100.16, 192.5.87.157	baremetal	uc-mc4n-key	Active	nova	None	Running	2 days, 3 hours	Attach Interface
Faulty node	CC-Ubuntu16.04	192.168.100.18, 192.5.87.38, 192.168.100.13	baremetal	uc-mc4n-key	Active	nova	None	Running	6 days, 4 hours	Attach Interface
Grafana-node	CC-Ubuntu16.04	192.168.100.19, 192.5.87.87	baremetal	uc-mc4n-key	Active	nova	None	Running	6 days, 20 hours	Attach Interface
Prometheus-node	CC-Ubuntu16.04	192.168.100.11, 192.5.87.178	baremetal	uc-mc4n-key	Active	nova	None	Running	6 days, 21 hours	Attach Interface
SRC-1-node	CC-Ubuntu16.04	192.168.100.17, 192.168.100.15, 192.5.87.186	baremetal	uc-mc4n-key	Shutoff	nova	None	Shut Down	6 days, 21 hours	Start Instance

Displaying 5 items

Challenge:
RL is promising to other Sciences too

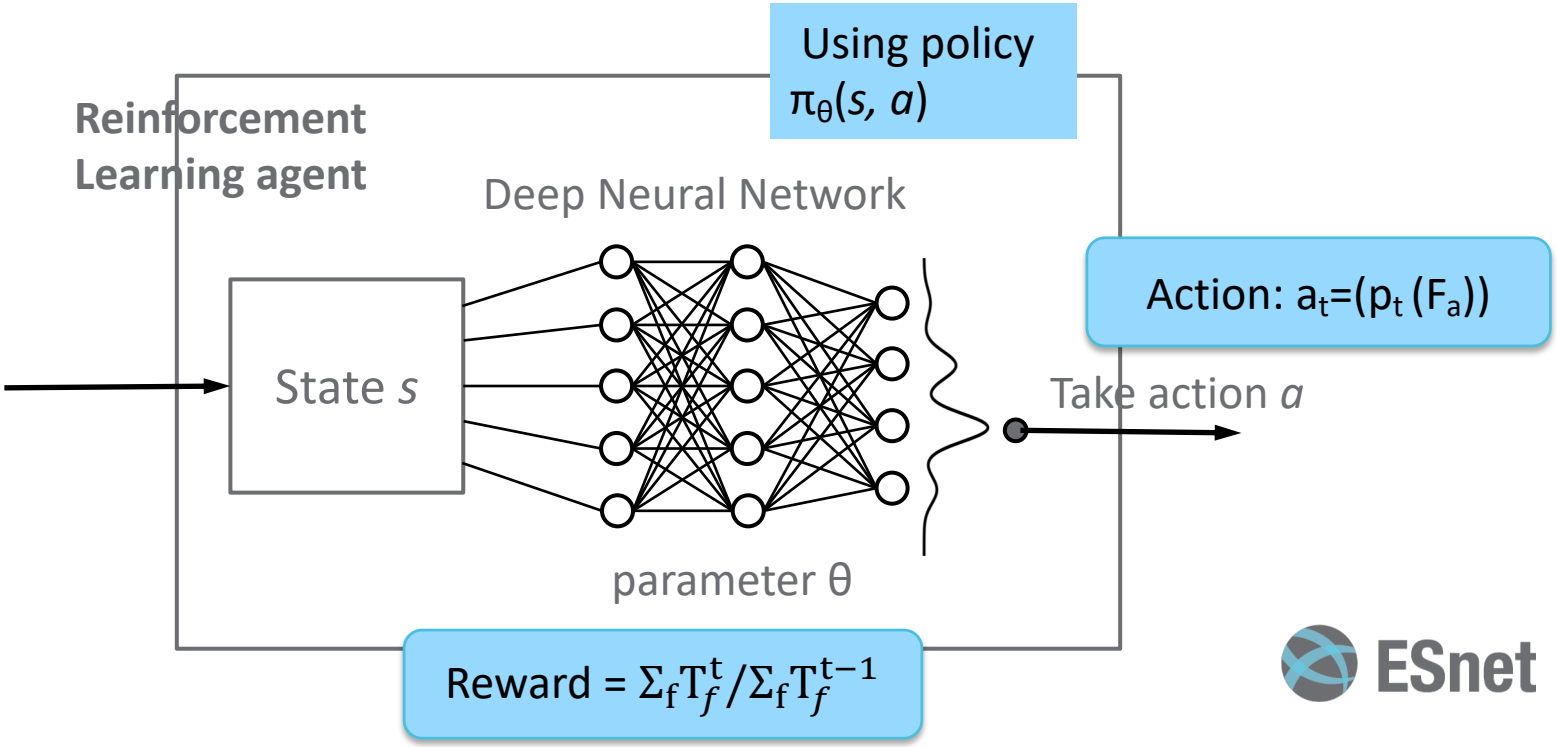
DQN (RL) for routing traffic

- Optimization problems in dynamic systems
- Not use Djikstra's shortest possible route finding
- HPC load balancing



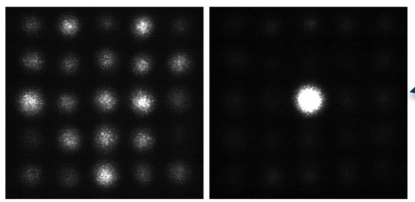
DRL applicable in many areas

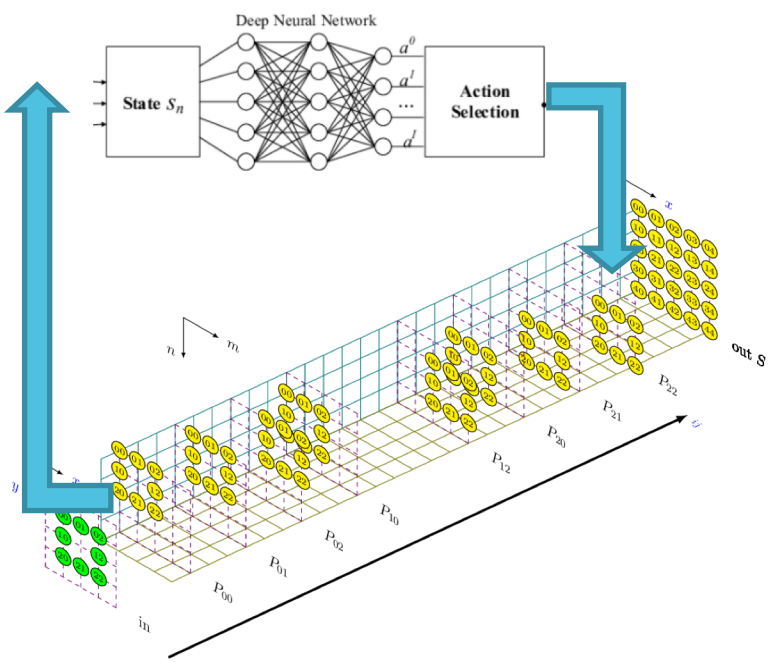
Replace the state and action to challenges in other dynamic systems



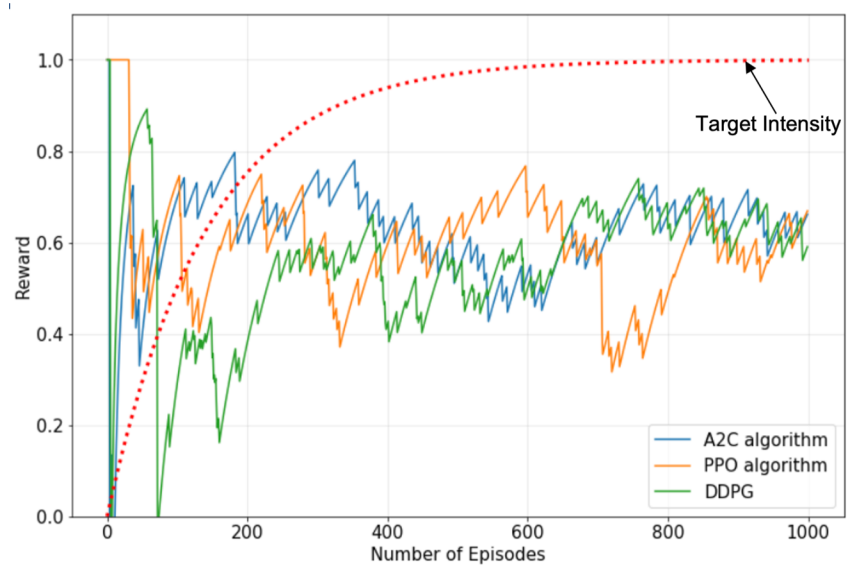
* Q. Du (ATAP)

DRL for 8-way Laser combining



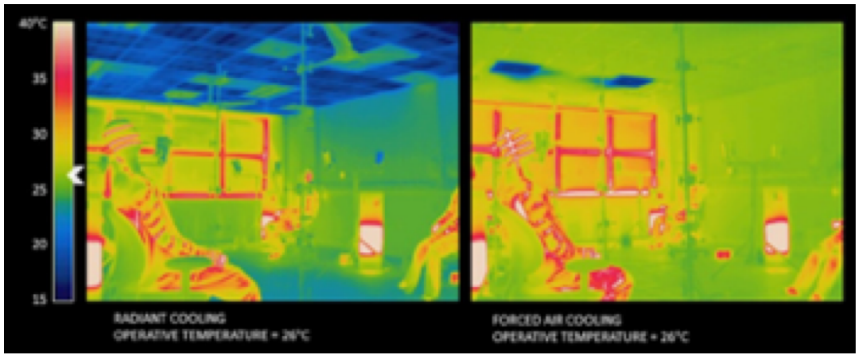


- System self-learns to find a concentrated beam
- Comparing RL approaches for real-time control
- Working with physicists to deploy DRL on GPUs/FPGAs for high-speed control

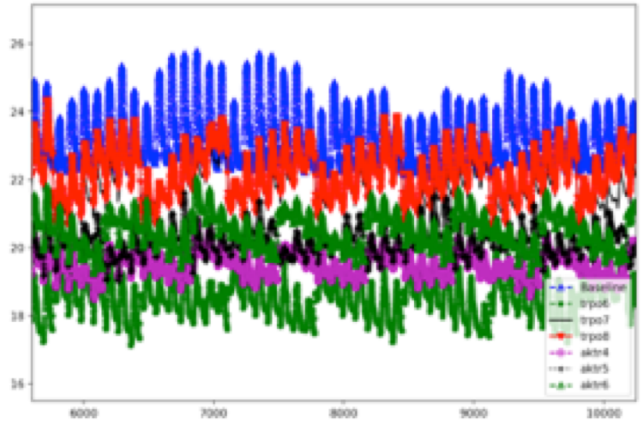


* S. Touzani (ETA)

DRL for FLEXLAB*



- DRL to optimize battery life and building comfort
- Hyperparameter tuning for real Flexlab deploys for temperature and cooling chambers
- Working with Energy technologists and experts



Temperature variability with different algorithms

ESnet

Summary:
Building our own Monitors/Controllers for optimizing traffic engineering

- Deploying algorithms to work with real-time data
- Building our own Cloud data management systems that communicate with our simulators to train offline and online models
- We have learned RL is a strong approach, but needs tweaking to problem to be beneficial that traditional approaches
- Working with Engineers/Domain scientists:
  - Deploying controllers in testbeds
  - Cost of processing data, training models and model reliability

ESnet

DAPHNE Team* *(Deep and Autonomous High-Performance Networks)*






* With ESnet engineers and SDM Collaborators

https://sites.google.com/lbl.gov/daphne

- mkiran@es.net

Feel free to reach out with questions/interests!