DeepConf: Automating Data Center Network Topologies Management with Machine Learning

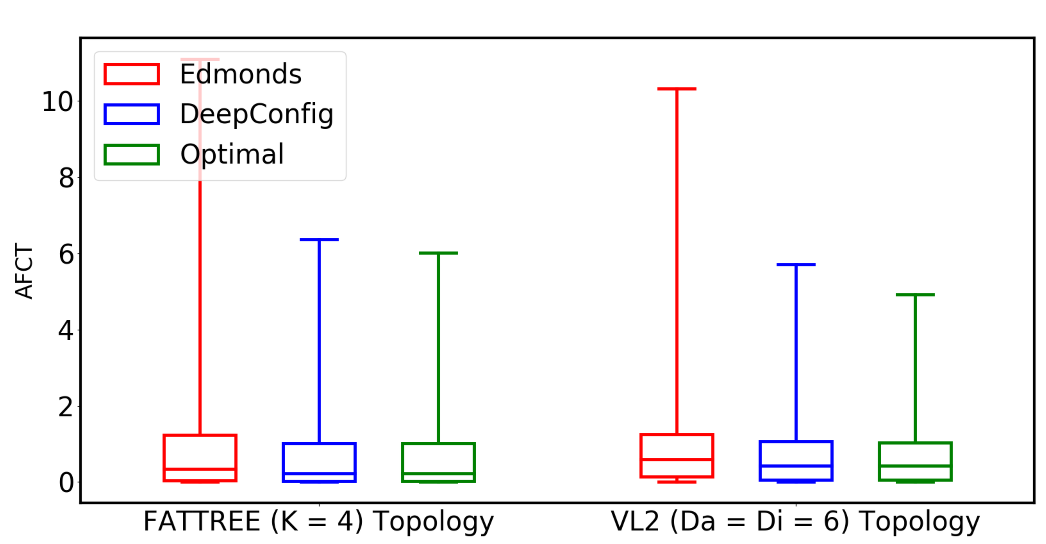
Saim Salman*, Theophilus Benson*, Asim Kadav‡
Brown University*, NEC Labs‡

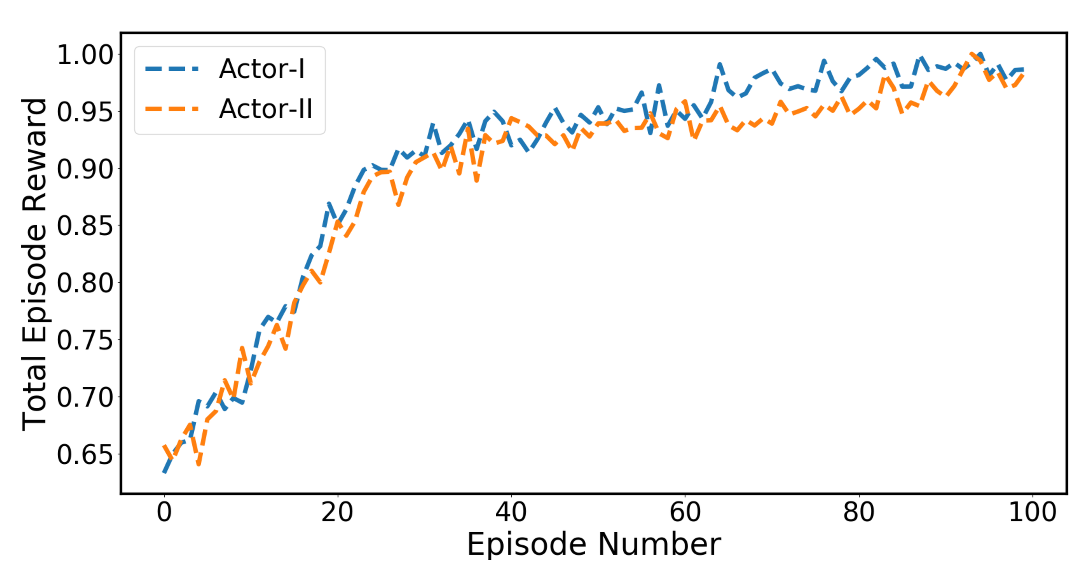
Problem

The networking community has explored and developed techniques (traffic engineering, topology augmentation) to improve the performance and efficiency of DC networks. But due to the complexity of these techniques (NP-Hard), greedy heuristics are employed to create approximate solution. But as these heuristics are intricately tied to the high-level application patterns and technological constraints the heuristics do not generalize. ***We present a system using reinforcement learning that captures intermediate representation of DC topologies to solve different classes of DC problems.***

Evaluation

- **Experiment Setup:** We evaluate DeepConf on a trace driven flow-level simulator using large scale map-reduce traces from Facebook. We evaluate two state-of-the-art clos-style data center topologies: K=4 Fat-tree and Da=Di=6 VL2. In our analysis, we focus on flow completion time (FCT) a metric which captures the duration between the first and last packet of a flow. We augment both topologies by adding an optical switch with five links. We compare DeepConf against: Optimal, the optimal solution derived from a brute-force program with perfect knowledge of future application demands and Edmonds a graph-based heuristic used in several existing works.

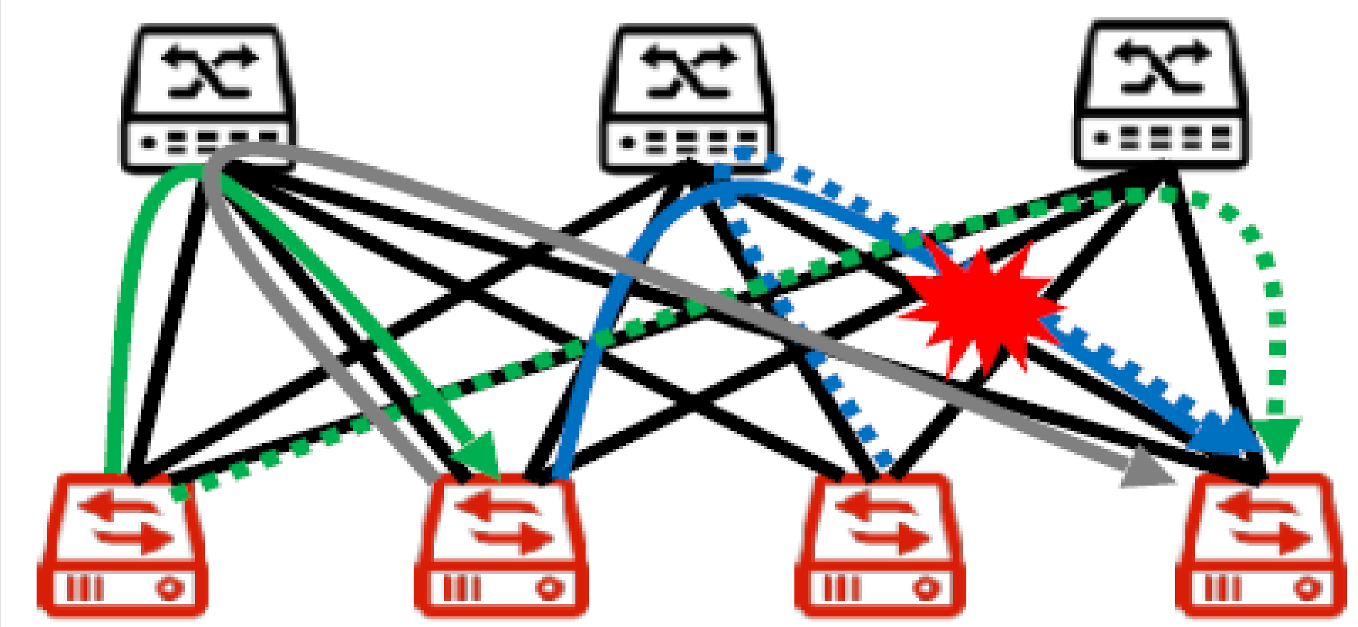




- **Model Learning:** The right figure presents the performance of each independent agent and we observe that each agent independently maximizes the total reward for each episode. The training results demonstrate that the RL agent learns to optimize its policy decision to increase the total reward received across each episode.
- **DeepConf Performance:** The left figure shows that DeepConf is able to learn a solution that's close to the optimal across representative data center topologies while outperforming Edmonds, a greedy heuristic used by many topology augmentation paper in terms of FCT.

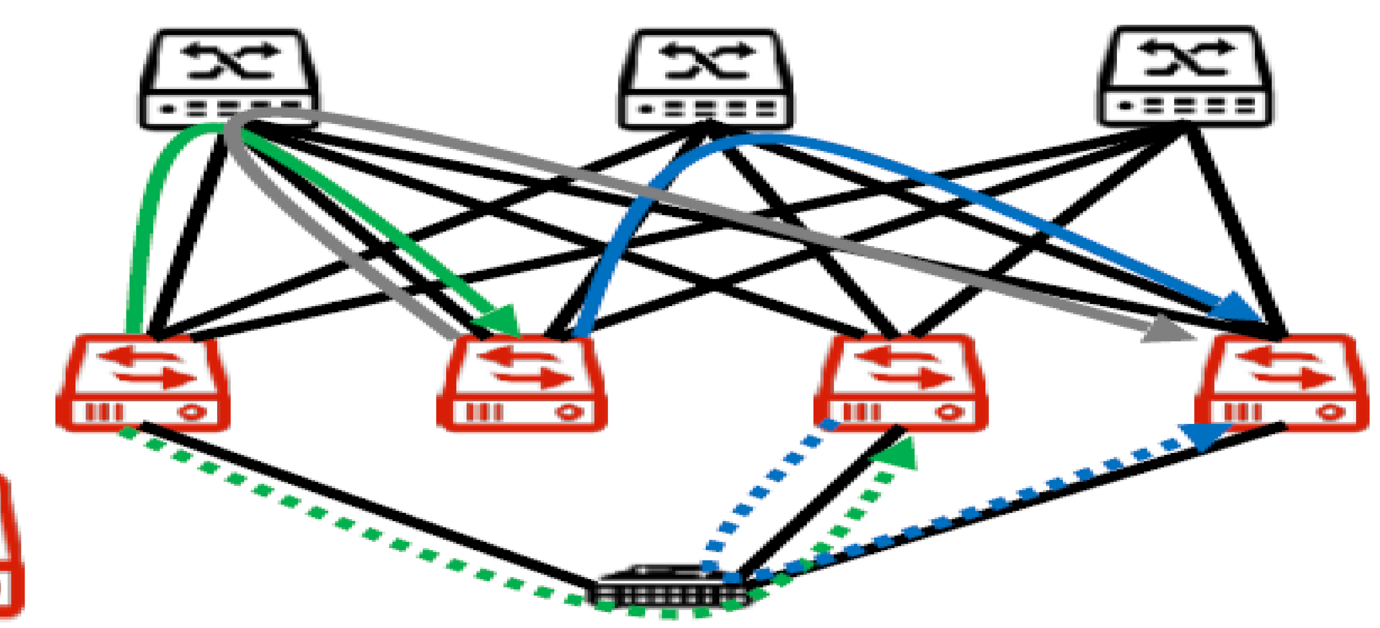
Data Center Topology Problems

Traffic Engineering (Routing)



Action Space: Paths in the network
Goal: Map flows to different paths

Topology Augmentation (Adding Links)

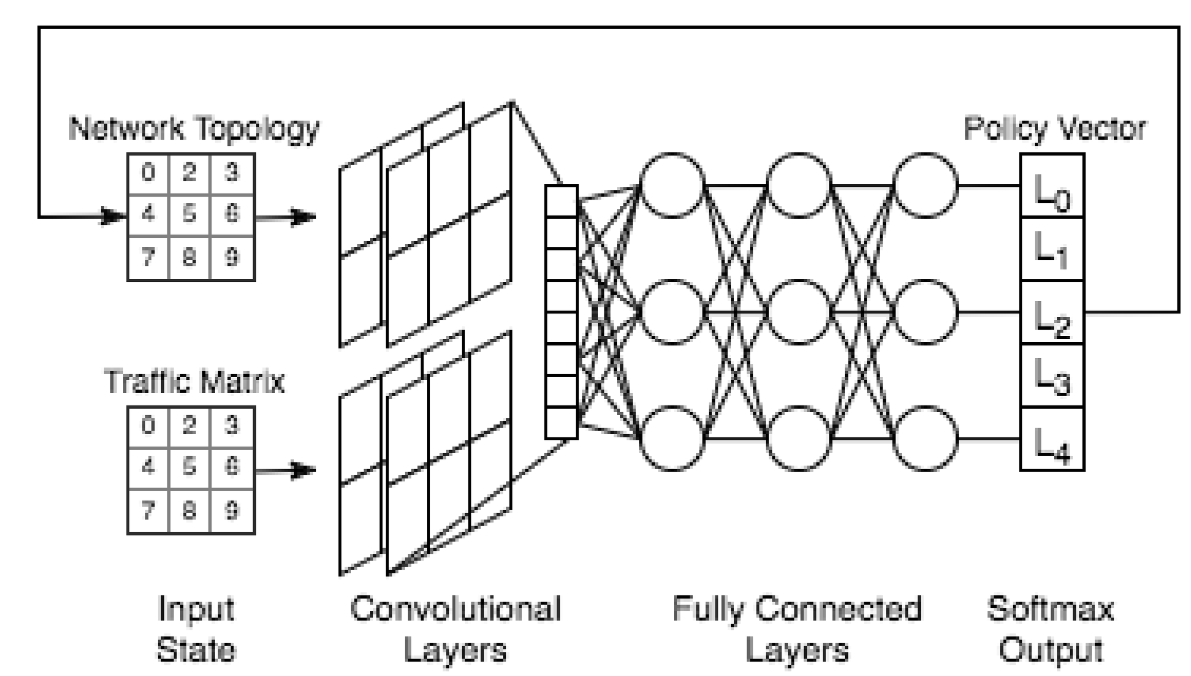


Action Space: Add k links
Goal: Identify locations to add links

Objective: **Minimize Average Flow Completion Time**

Topology Augmentation

- **Input State**
  - **Network Topology:** NxDxD Matrix where D is the number of ToRs in the network and N is the batch size.
  - **Traffic Matrix:** NxDxDx2 Matrix where (I,J) th entry represent flow information between ToR_I and ToR_J.



Input-Space	CNN-1	CNN-2	Fully Connected	Value Network	Policy Network
Traffic Matrix (NxDxDx2)	Nx5x5x(D*2)	Nx2x2x(D*4)	Nx256	Nx256*K	Nx256xO
Topology (NxDxD)	Nx5x5x(D*2)	Nx2x2x(D*4)			

- The **Neural Network** consists of two CNN layers followed by a fully connected layer.
- The **Policy Networks'** output (O) represents the number of potential optical links from which K are selected.
- Our **Reward Function** is a K sized vector where each index presents the bytes traversed over that specific optical link.
- The policy affects the network topology directly and the actions update the environment which updates the traffic matrix.