Deep Reinforcement Learning-Based Routing on Software-Defined Networks

Professor: Li-Der Chou

Presenter: Chih-Jou Tai

IEEE Access Feb 2022

Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

Introduction

- Traditional Distributed Routing
  - Independence
  - Difficulty in Global Optimization
- Software-Defined Networking (SDN)
  - Centralized Management
  - Flexibility and Optimization

Introduction

- Utilizing DRL in SDN
  - Optimization and adaptability
  - Learning delay and instability
- Solutions
  - DDPG (Deep Deterministic Policy Gradient)
  - ATVM (Aggregated Traffic Volume Matrix)
  - M/M/1/K Queue Model

Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

Related Work

- Traditional Routing Methods:
  - OSPF
  - ECMP
- SDN-Based Routing Methods:
  - Fully Polynomial Time Approximation Scheme.
  - Simulated Annealing QoS Routing for SDN-based IoT.

Related Work

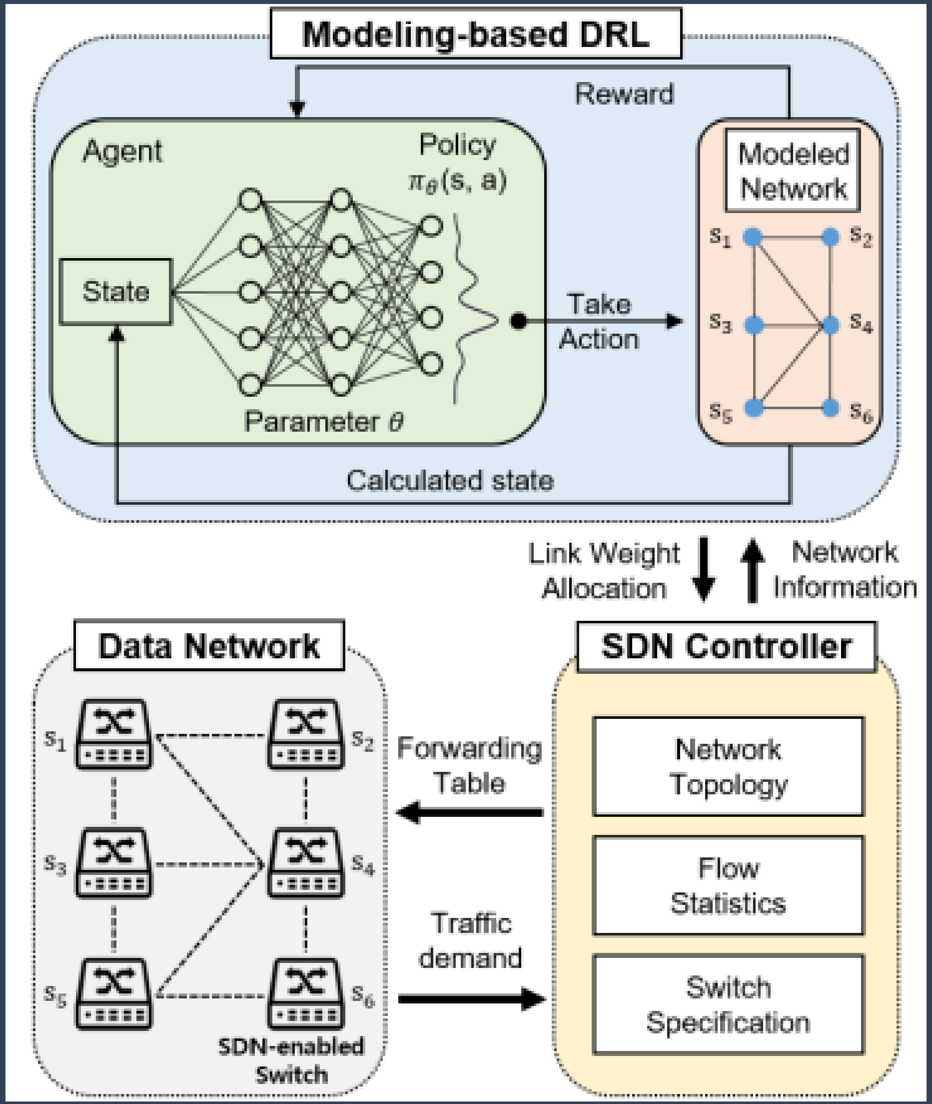
- Machine Learning-Based Routing Methods:
  - Supervised Deep Learning for Routing Table Construction.
  - Deep Q-learning-based Routing Strategy for Data Center Networks.

Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

System Model

- Overall System Architecture

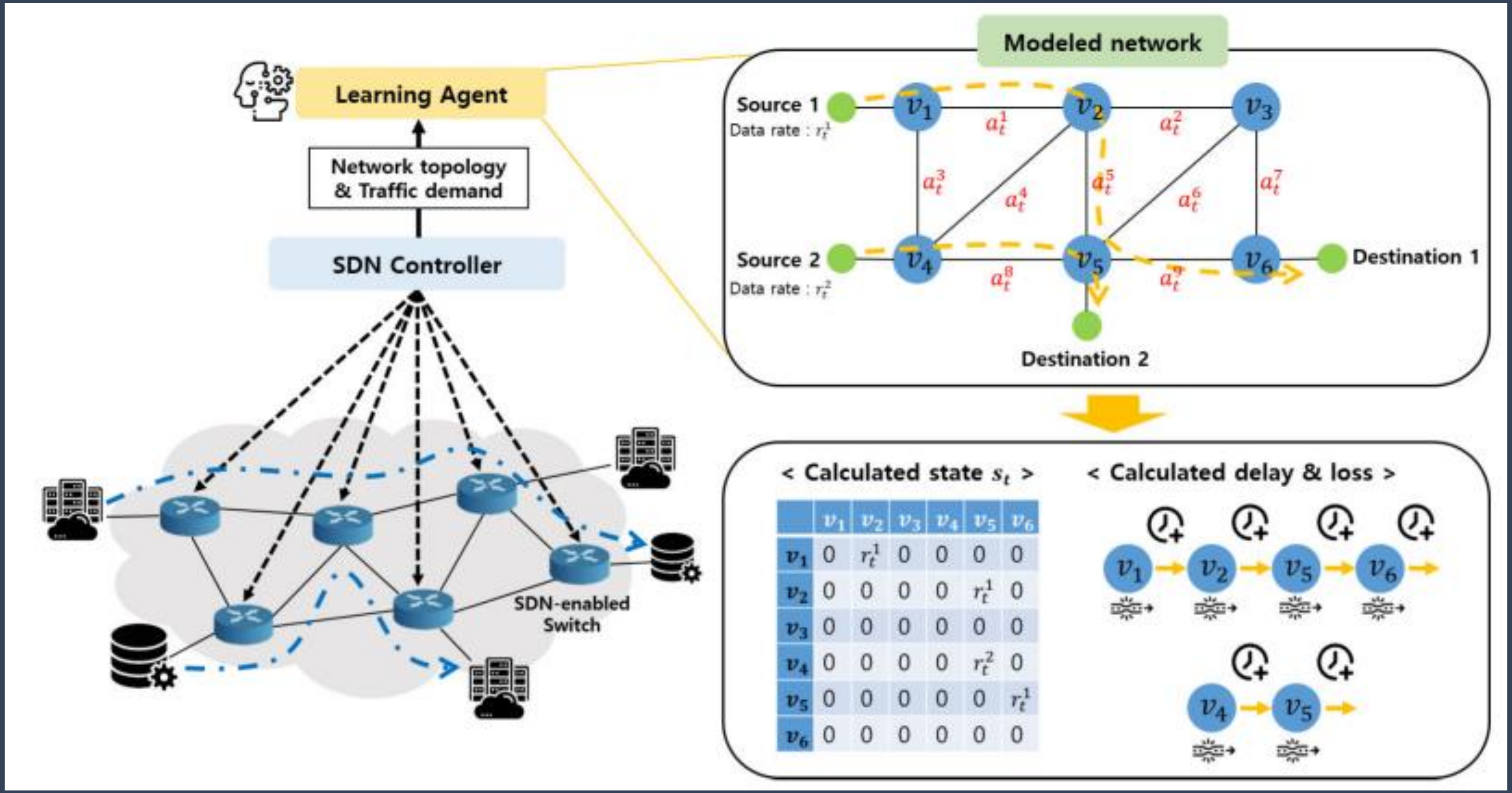


System Model

- Training takes place in a simulated environment.
- Training speed is not constrained by real-time but depends on hardware computational capacity.
- DDPG algorithm is employed to update neural networks for optimizing weight allocation.

System Model

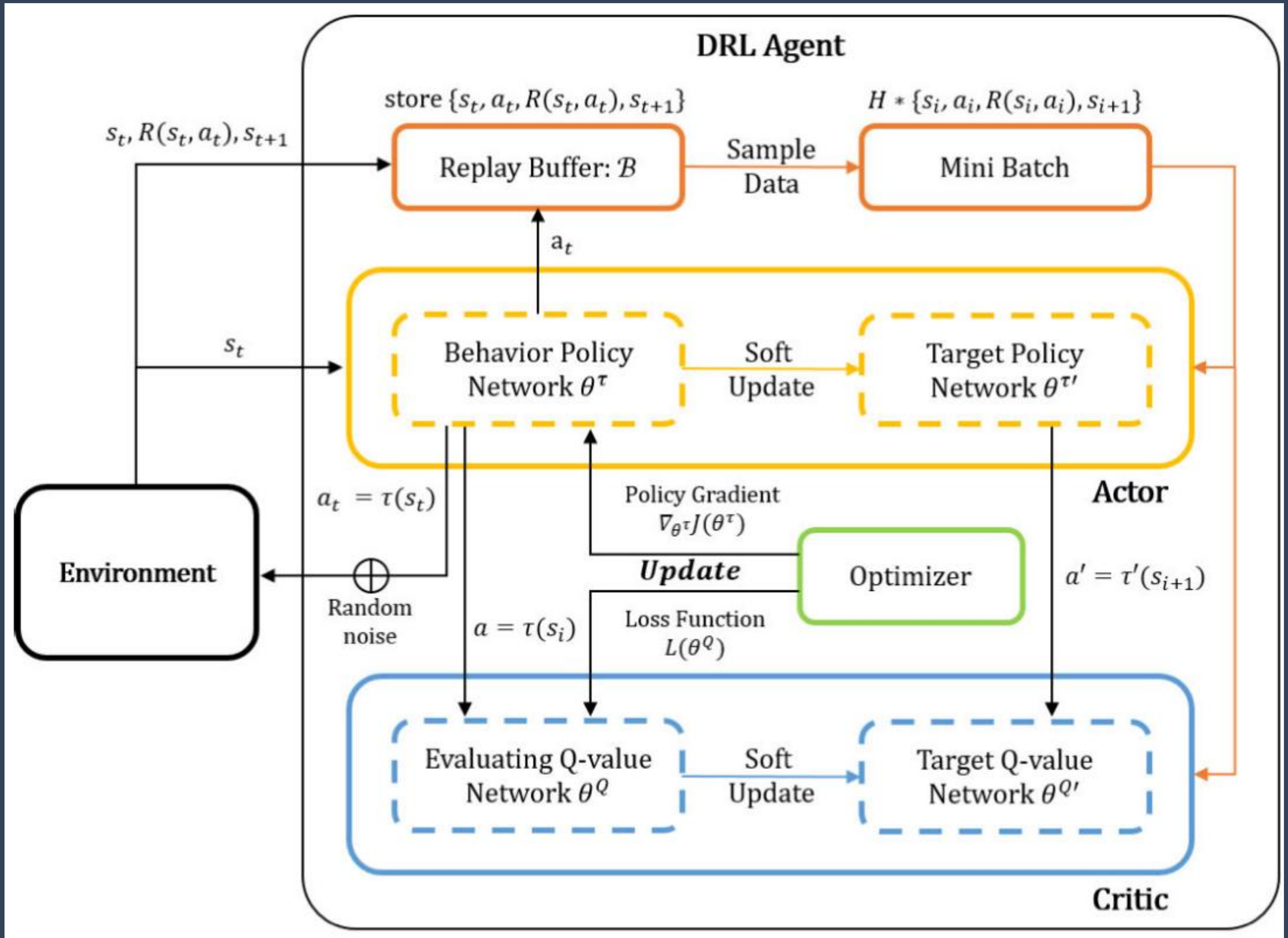
ATVM(Aggregated Traffic Volume Matrix)



	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	r_t^1	0	0	0	0
v_2	0	0	0	0	r_t^1	0
v_3	0	0	0	0	0	0
v_4	0	0	0	0	r_t^2	0
v_5	0	0	0	0	0	r_t^1
v_6	0	0	0	0	0	0

System Model

DDPG training process diagram



Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

Algorithm

- Initialization
  - critic-network $Q(s, a|\theta^Q)$
  - actor-network τ $(s|\theta^\tau)$
  - replay buffer B
  - M/M/1/K model
  - Initial state s0
- Repeat
  - Select action and calculate R(st, at), st+1
  - Store {st, at, R(st, at), st+1} in B
  - A batch of H transtions{si, ai, R(si, ai), si+1}
  - Update θ^Q and θ^τ with optimizer

Algorithm 1 The Proposed DDPG-Based Routing Algorithm

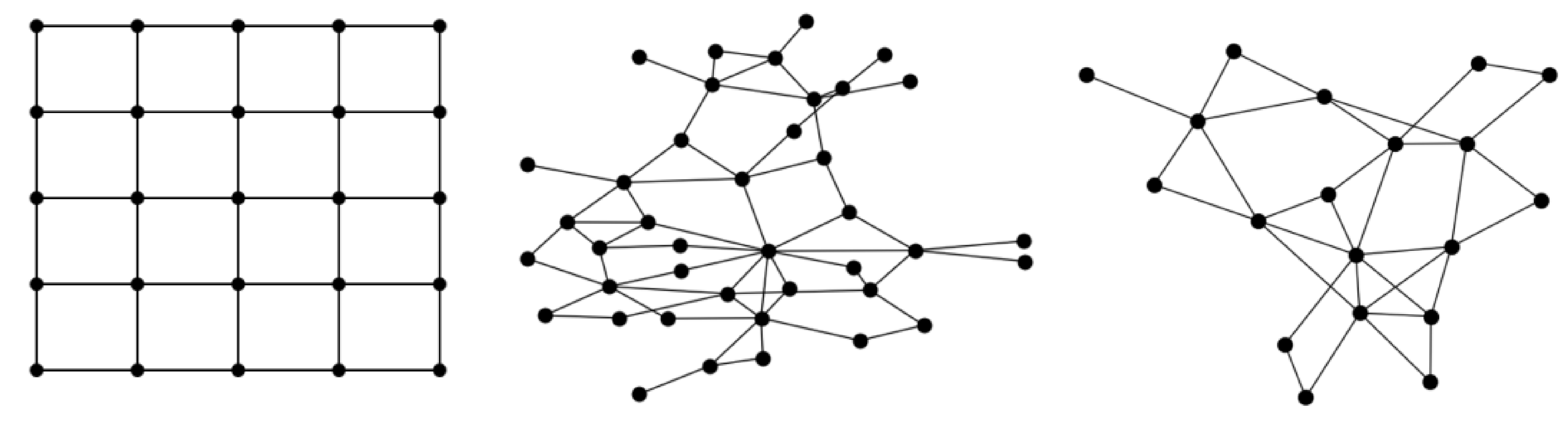
// Initialization
Set the critic-network $Q(s, a|\theta^Q)$ and actor-network $\tau(s|\theta^\tau)$ with randomly generated weight θ^Q and θ^τ
Set target parameters equal to main parameters $\theta^{Q'} \leftarrow \theta^Q, \theta^{\tau'} \leftarrow \theta^\tau$
Empty experience replay buffer $\mathcal{B}$
Construct the M/M/1/K queue-based network model using network information from SDN controller.
Set initial state s_0 in accordance with the initial routing policy
// Parameter updating
repeat
    Select action $a_t = \tau(s_t|\theta^\tau) + \mathcal{N}$ following the parameter noise for exploration
    Take action a_t on modeled network and calculate $R(s_t, a_t)$, s_{t+1}
    Store transition $\{s_t, a_t, R(s_t, a_t), s_{t+1}\}$ in $\mathcal{B}$
    if it's time to update **then**
        Update the network information from SDN controller.
    end if
    Randomly sample a batch of H transitions $\{s_i, a_i, R(s_i, a_i), s_{i+1}\}$ from $\mathcal{B}$
    Set $y_i = R(s_i, a_i) + \gamma Q'(s_{i+1}, \tau'(s_{i+1}|\theta^{\tau'})|\theta^Q)$
    Update critic θ^Q and actor θ^τ in (18) and (19)
    Update the targets softly in (20) and (21)
until convergence

Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

Performance Evaluation

- Network Topologies
  - Grid Topology
  - GEANT Topology
  - InternetMCI Topology



(a) Grid topology with 25 switches (b) GEANT topology with 40 switches (c) InternetMCI topology with 19 switches

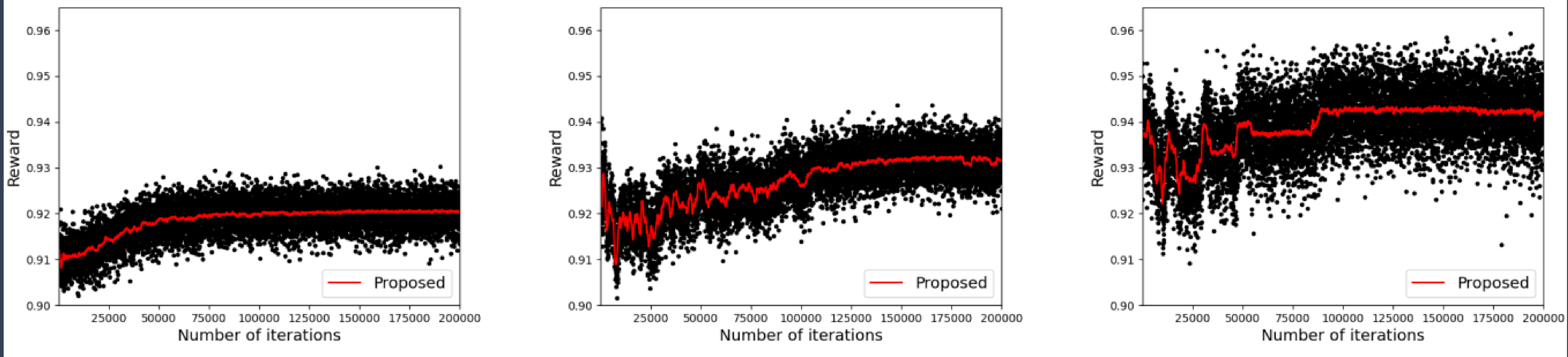
Performance Evaluation

- Simulation Settings
  - System capacity : 10,000 packets
  - Service rate : 3,000 packets/second.
  - Arrival rate : Between 10 and 300 packets/second
  - Link weights : Range from 1 to 5
  - DDPG Algorithm

Hyper-parameter	Value
Discount factor γ	0.99
Replay buffer $\mathcal{B}$	50,000
Batch size H	100
Critic learning rate ϵ_c	0.00001
Actor learning rate ϵ_a	0.00001

Performance Evaluation

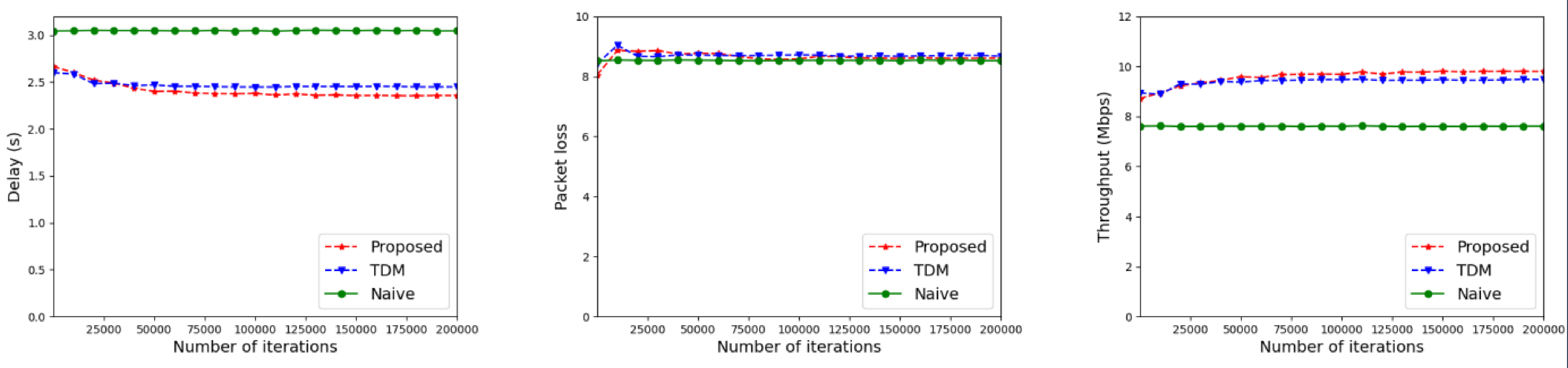
- Rewards of the agent with respect to the number of time steps.



(a) Grid topology (b) GEANT topology (c) InternetMCI topology

Performance Evaluation

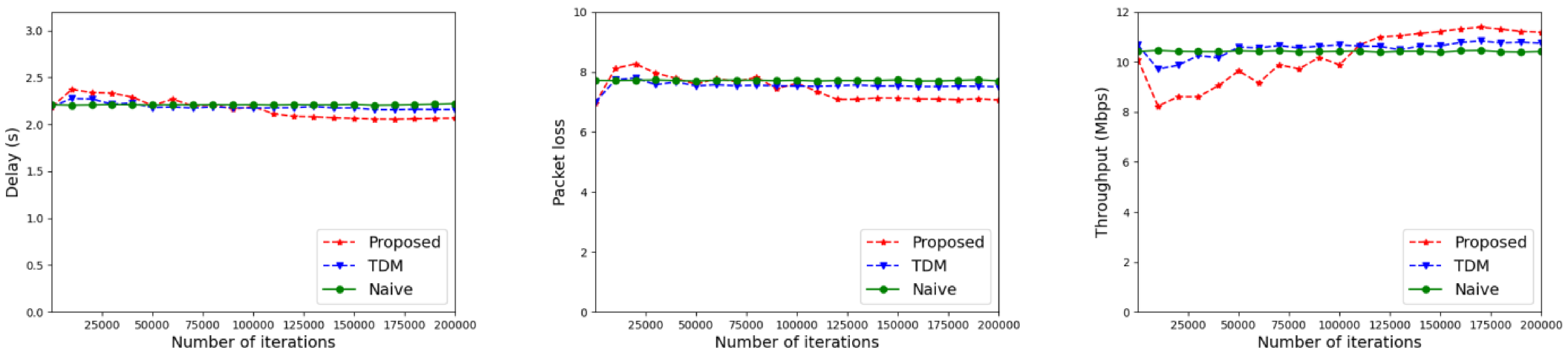
- Average network performance with respect to the number of iterations in grid topology.



(a) Delay (b) Packet loss (c) Throughput

Performance Evaluation

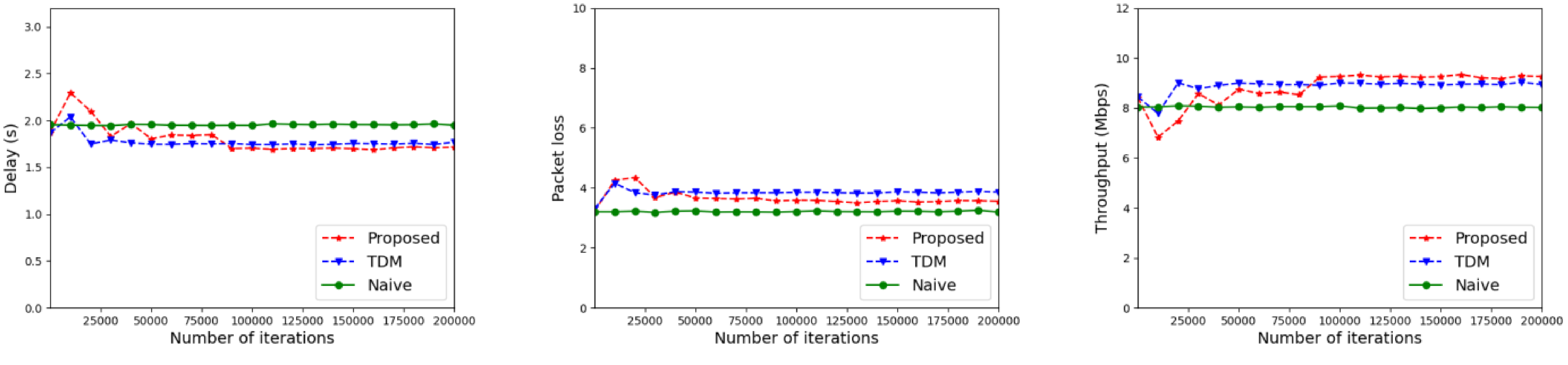
- Average network performance with respect to the number of iterations in GEANT topology.



(a) Delay (b) Packet loss (c) Throughput

Performance Evaluation

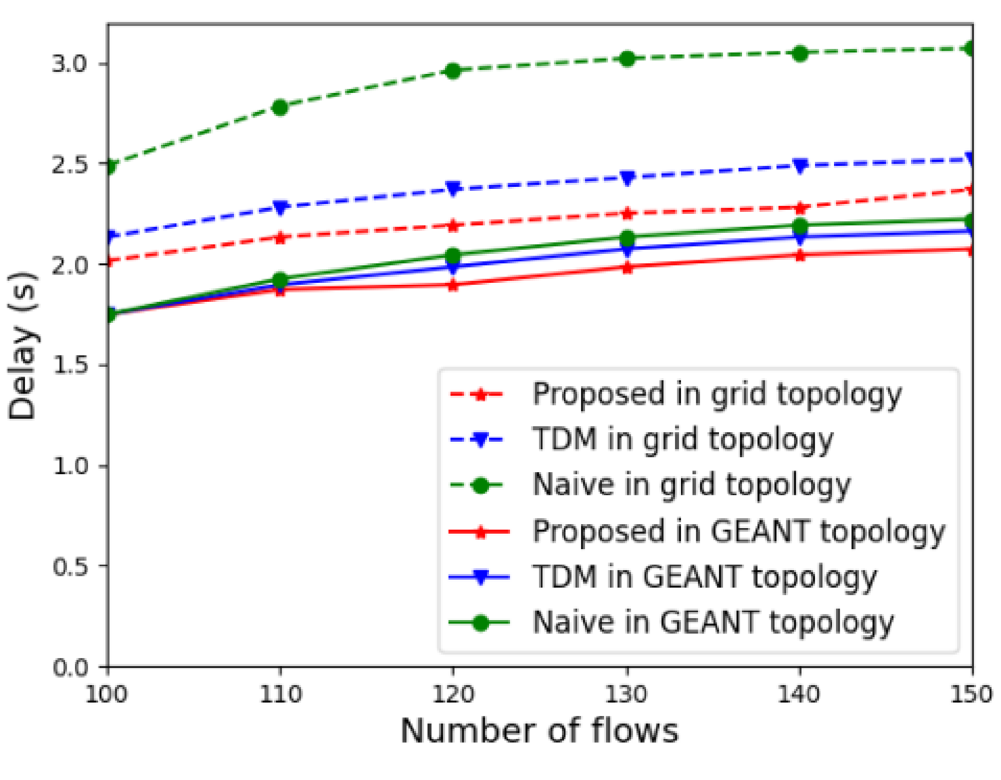
- Average network performance with respect to the number of iterations in InternetMCI topology.



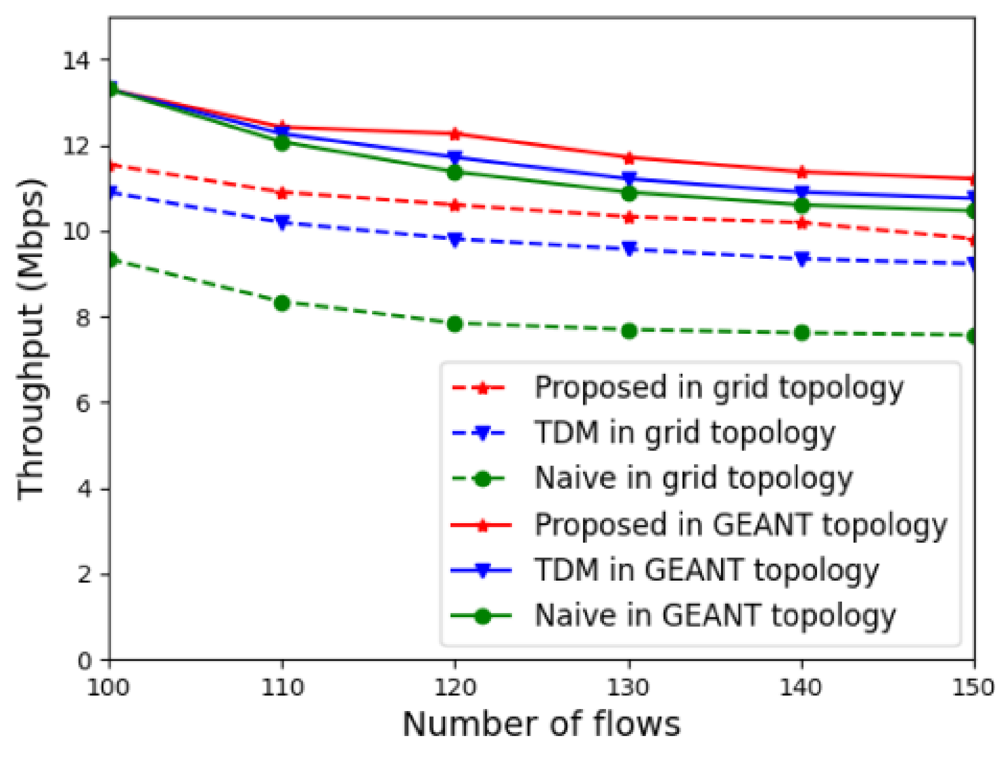
(a) Delay (b) Packet loss (c) Throughput

Performance Evaluation

- Average network performance with respect to the number of flows.



(a) Delay with respect to the number of flows



(b) Throughput with respect to the number of flows

Outline

- Introduction
- Related Work
- System Model
- Algorithm
- Performance Evaluation
- Conclusion

Conclusion

- Pros:
  - A routing strategy that combines SDN and DRL is proposed, which may offer higher network performance than traditional methods.
  - The use of the M/M/1/K model to simulate the network reduces the risk and cost of experiments in real network environments.
- Cons:
  - In environments with highly dynamic topology changes (Wireless Networks, Internet of Things), a new model must be trained with each change.
- Future works:
  - Adopting graph neural networks that generate a generalized model applicable across multiple topologies can enhance computational efficiency.

Q & A