Neural Packet Routing

Shihan Xiao, Haiyan Mao, Bo Wu, Wenjie Liu, Fenglin Li

Network Technology Lab,

Huawei Technologies Co., Ltd., Beijing, China

Motivation

Today's distributed routing protocols

- Advantage
  - Connectivity guarantee
- Disadvantage
  - 1. Difficult to be extended to satisfy flexible optimization goals
  - 2. Big time and human costs to design and tune the configurations to achieve the optimal

Future network expectations

- Flexible optimization goals beyond connectivity guarantee
  - 5G applications desire lowest end-to-end delay
  - Industry network applications require deterministic end-to-end delay
- Less human costs to achieve the optimal
  - Future network is expected to be highly automated with less and less human costs

Motivation

Today's distributed routing protocols

- Advantage
  - Connectivity guarantee
- Disadvantage
  - 1. Difficult to be extended to satisfy flexible optimization goals
  - 2. Big time and human costs to design and tune the configurations to achieve the optimal

Future network expectations

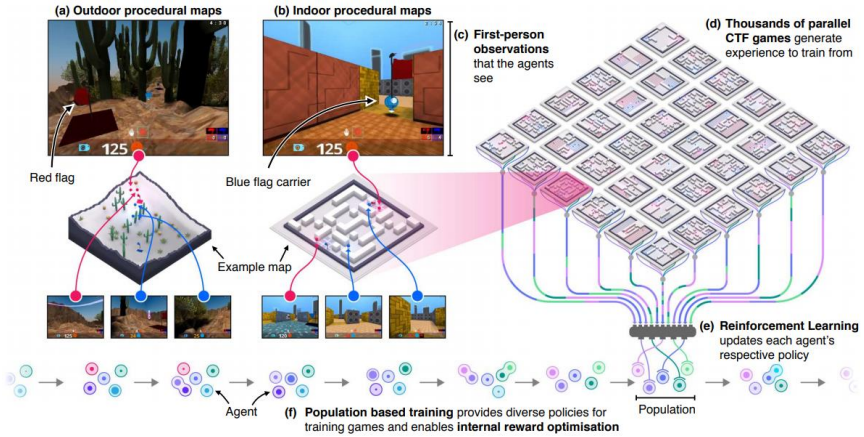
- Flexible optimization goals beyond connectivity guarantee
  - 5G applications desire lowest end-to-end delay
  - Industry network applications require deterministic end-to-end delay
- Less human costs to achieve the optimal
  - Future network is expected to be highly automated with less and less human costs

Can we achieve the flexible and automated optimal protocol design at the same time?

Motivation

- Deep learning can be seen as one potential way for achieving both the flexible and automated optimality of distributed routing

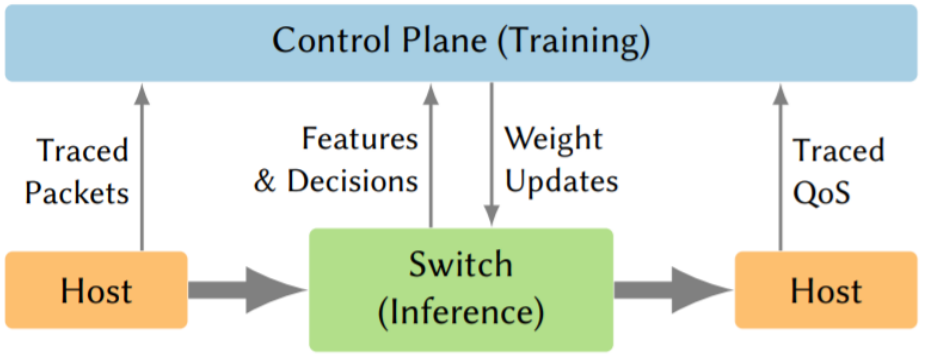
Deep learning in multi-agent game surpass human performance



[DeepMind, 2018]

Figure source: https://arxiv.org/abs/1807.01281

Line-rate neural network inference in future switches



[Swamy et al., 2020]

Figure source: https://arxiv.org/abs/2002.08987

Motivation

- Deep learning is a good start
- A simple learning-based distributed routing framework

- **Key idea: train a deep neural network at each node (router/switch) to compute the forwarding port for each packet**

Motivation

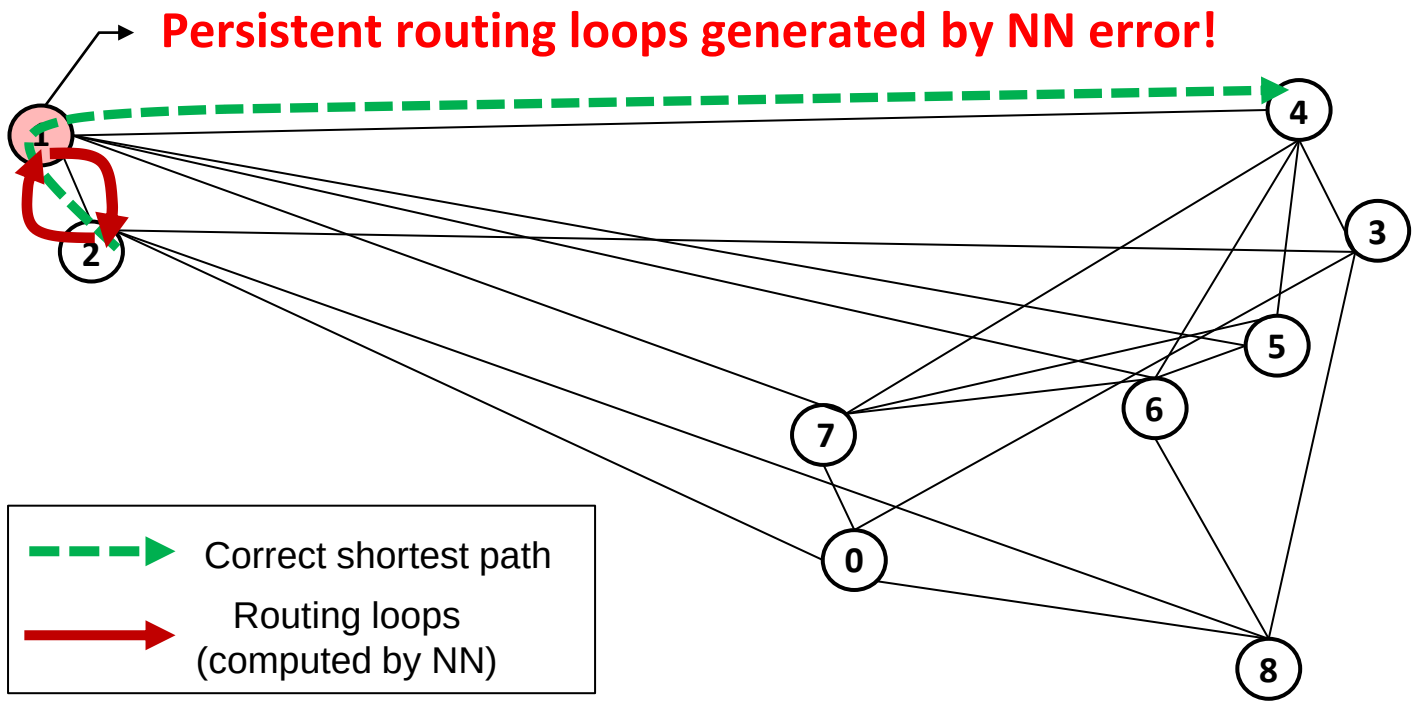
- Deep learning is a good start
- A simple learning-based distributed routing framework

- **Key idea: train a deep neural network at each node (router/switch) to compute the forwarding port for each packet**

Question about the "learning safety": What will happen if neural network makes mistakes?

Motivation

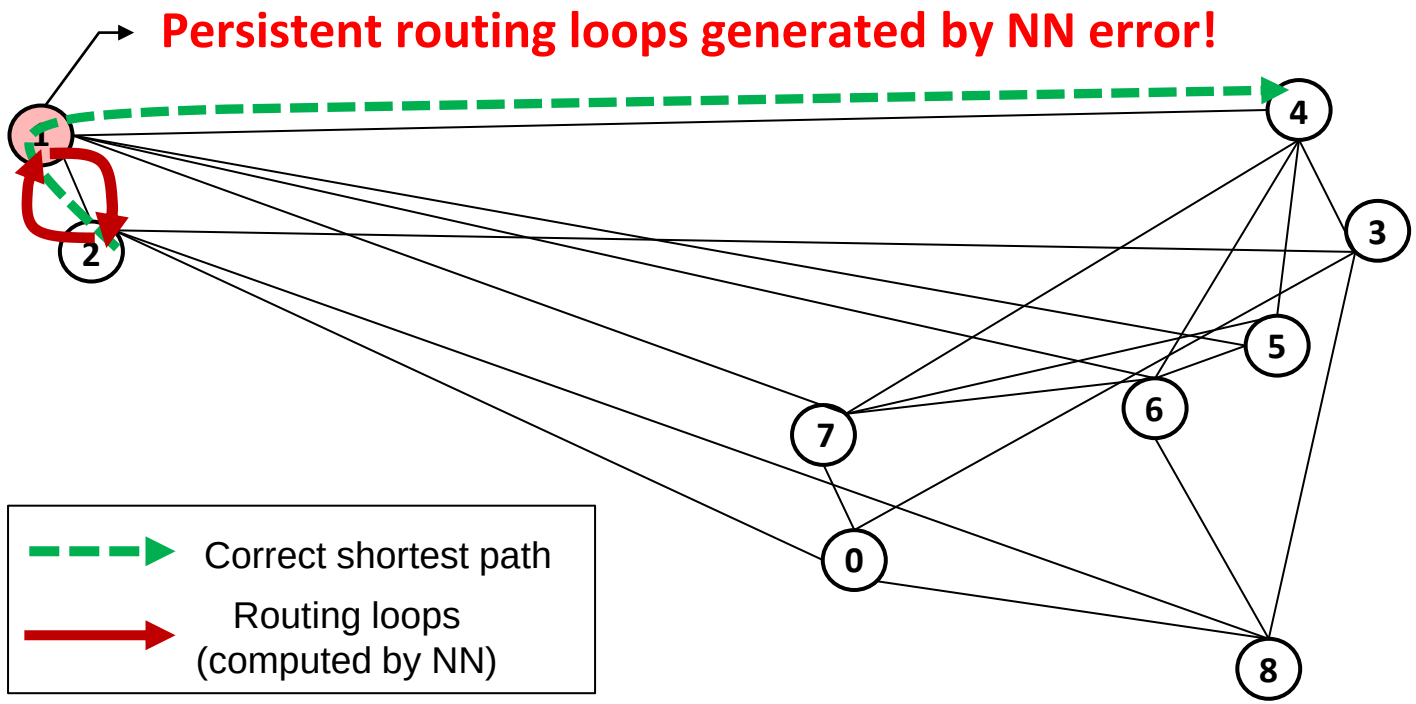
- Deep learning is a good start, **but there is still a reality "gap"**
- A simple learning-based distributed routing framework



Simulation of shortest-path supervised learning with 97.5% training accuracy

Motivation

- Deep learning is a good start, **but there is still a reality "gap"**
- A simple learning-based distributed routing framework



The inference error in deep learning is unavoidable. Can we still achieve reliability guarantee while keeping the advantages of deep learning?

Solution: Neural Guided Routing (NGR)

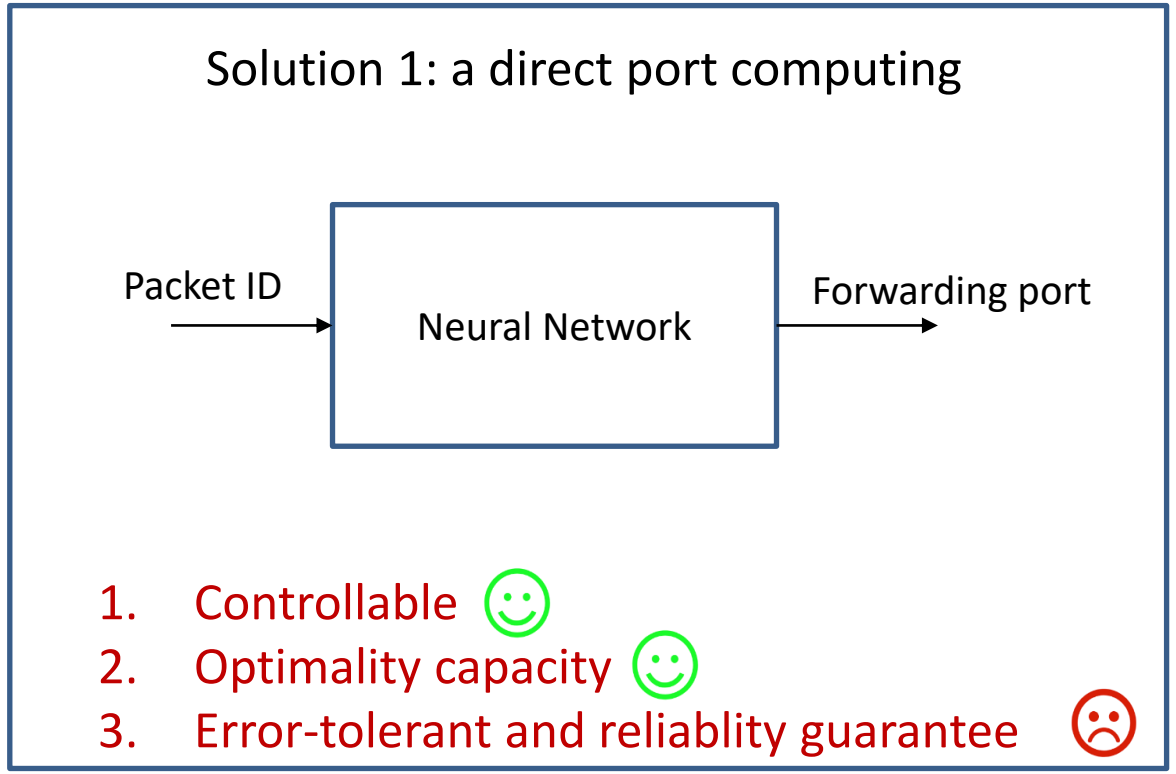
- Overview of NGR design:
  - 1. A reliable distributed routing framework
  - 2. Combine deep learning into the framework
  - 3. Handle the topology changes

Solution: Neural Guided Routing (NGR)

- A reliable distributed routing framework
  - We define a routing path is *reliable* if it reaches the destination without any persistent loops/blackholes
- Desired properties of the routing framework
  - 1. Controllable:
    - It has some parameters W to directly control the routing path for each packet
  - 2. Optimality capacity:
    - Any reliable routing path can be implement by setting proper W
  - 3. Error-tolerant and reliability guarantee:
    - It always generates a reliable routing path no matter what errors happen in setting W

Solution: Neural Guided Routing (NGR)

- The challenge in finding such a routing framework



Solution: Neural Guided Routing (NGR)

- The challenge in finding such a routing framework

Solution 1: a direct port computing

1. Controllable ☺
2. Optimality capacity ☺
3. Error-tolerant and reliability guarantee ☹

Solution 2: triangle-constraint routing

Triangle constraint: only use neighboring nodes that are closer to the destination as the next hop

Triangle-constraint routing ⟷ Optimal routing

Optimality Gap

1. Controllable ☺
2. Optimality capacity ☹
3. Error-tolerant and reliability guarantee ☺

Solution: Neural Guided Routing (NGR)

- The challenge in finding such a routing framework

Solution 1: a direct port computing

1. Controllable ☺
2. Optimality capacity ☺
3. Error-tolerant and reliability guarantee ☹

Solution 2: triangle-constraint routing

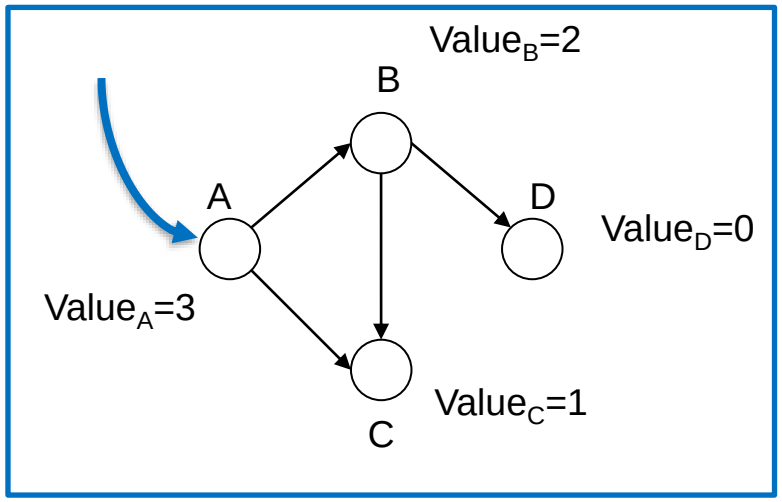
Triangle constraint: only use neighboring nodes that are closer to the destination as the next hop

1. Controllable ☺
2. Optimality capacity ☹
3. Error-tolerant and reliability guarantee ☺

Key Question: Can we find a framework satisfying all the desired properties?

Solution: Neural Guided Routing (NGR)

- NGR proposes new routing framework S-LRR following the link reversal theory
  - Key idea: assign a value to each node, and link directions are defined from high-value node to low-value node; add updated node value into packet head to implement the link reversal



Workflow example:
A packet with destination D arrives at node A

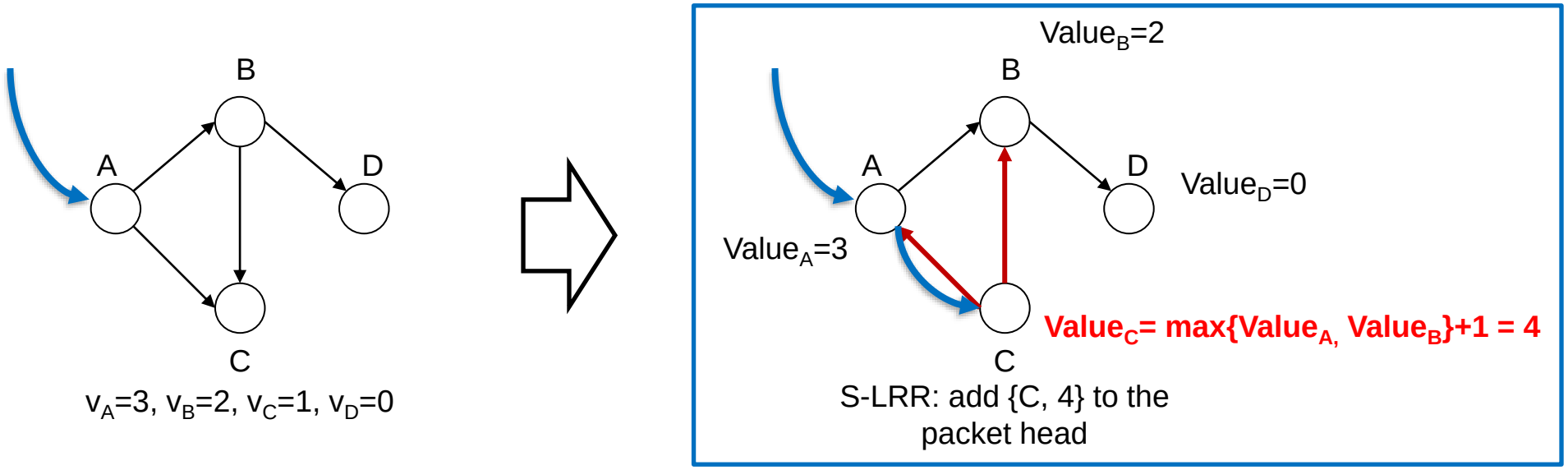
- **Forwarding rule 1:** the next-hop node can only be selected **from lower-value neighboring nodes**; when there are multiple choices of next-hop nodes, **select the one with lowest value**

⇩

Next-hop node is C

Solution: Neural Guided Routing (NGR)

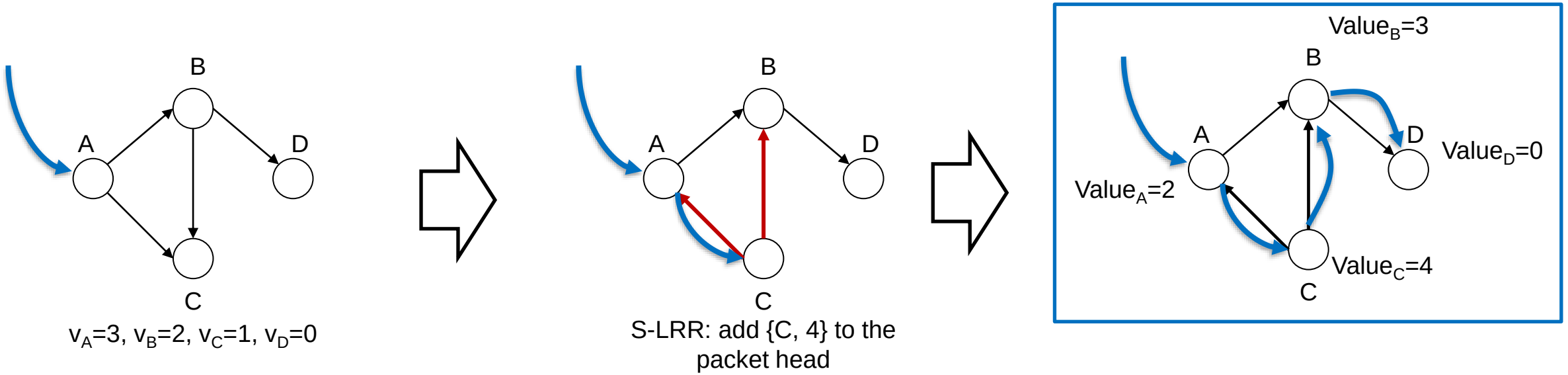
- NGR proposes new routing framework S-LRR following the link reversal theory
  - Key idea: assign a value to each node, and link directions are defined from high-value node to low-value node; add updated node value into packet head to implement the link reversal



- **Forwarding rule 2: if current node is a sink node (i.e., no out-links), perform the link reversal operation**
  --change the current node value to new Value = max{neighboring node values} + 1
  --add the key-value pair {current node index, new Value} into packet head
  --next-hop node will extract the packet head to get the newest node value

Solution: Neural Guided Routing (NGR)

- NGR proposes new routing framework S-LRR following the link reversal theory
  - Key idea: assign a value to each node, and link directions are defined from high-value node to low-value node; add updated node value into packet head to implement the link reversal

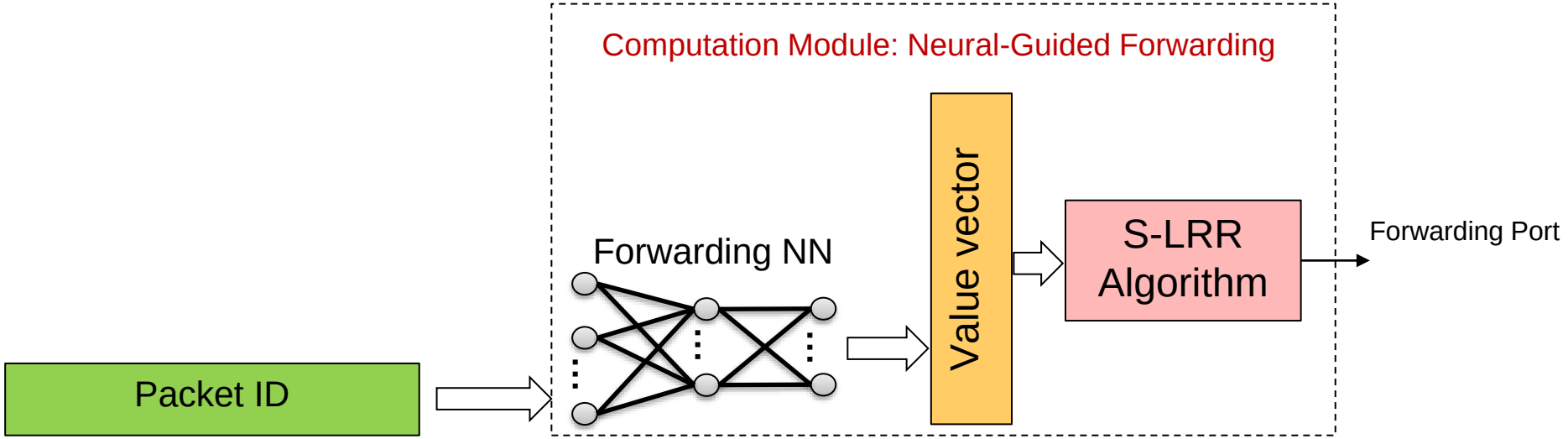


Repeat using the following rules until reach the destination (guaranteed by the link reversal theory):

- Forwarding rule 1: the next-hop node can only be selected from lower-value neighboring nodes; when there are multiple choices of next-hop nodes, select the one with lowest value
- Forwarding rule 2: if current node is a sink node, perform the link reversal operation

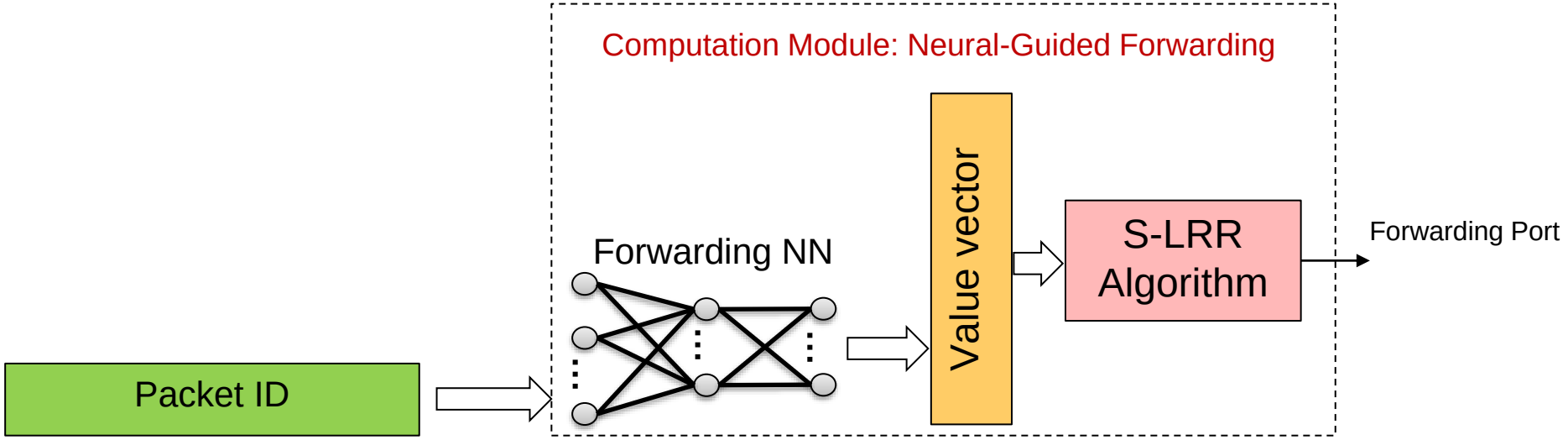
Solution: Neural Guided Routing (NGR)

- Combining deep learning into the routing framework
  - Key idea: the final routing path is controlled by the node values
  - NGR trains a neural network at each node to compute the node values, so that the neural network can learn to optimize the routing path directly



Solution: Neural Guided Routing (NGR)

- Combining deep learning into the routing framework
  - Key idea: the final routing path is controlled by the node values
  - NGR trains a neural network at each node to compute the node values, so that the neural network can learn to optimize the routing path directly



The above framework is
1) Controllable based on node values
2) Error-tolerant with reliability guarantee based on link reversal theory
But what about its optimality capacity?

Solution: Neural Guided Routing (NGR)

- Combining deep learning into the routing framework
  - To achieve the optimality capacity of the combined deep-learning framework, a fine-grained patch is required
    - Each node is assigned two values separately, termed the Prime value and Secondary value

- Forwarding rule 1: the next-hop node can only be selected **from lower-value neighboring nodes**; when there are multiple choices of next-hop nodes, **select the one with lowest value**
- Forwarding rule 2: if current node is a sink node, perform the link reversal operation

Prime value:
Decide the feasible set of next-hop nodes

Secondary value:
Decide the final next-hop selection

Solution: Neural Guided Routing (NGR)

- Combining deep learning into the routing framework
  - To achieve the optimality capacity of the combined deep-learning framework, a fine-grained patch is required
    - Each node is assigned two values separately, termed the Prime value and Secondary value

- Forwarding rule 1: the next-hop node can only be selected **from lower-value neighboring nodes**; when there are multiple choices of next-hop nodes, **select the one with lowest value**
- Forwarding rule 2: if current node is a sink node, perform the link reversal operation

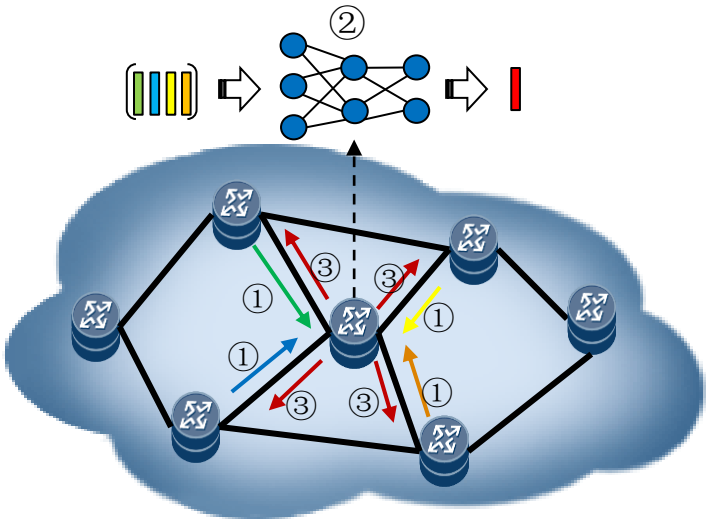
Prime value:
Decide the feasible set of next-hop nodes

Secondary value:
Decide the final next-hop selection

We prove that the above framework can achieve the reliability guarantee while keeping the optimality capacity of deep learning

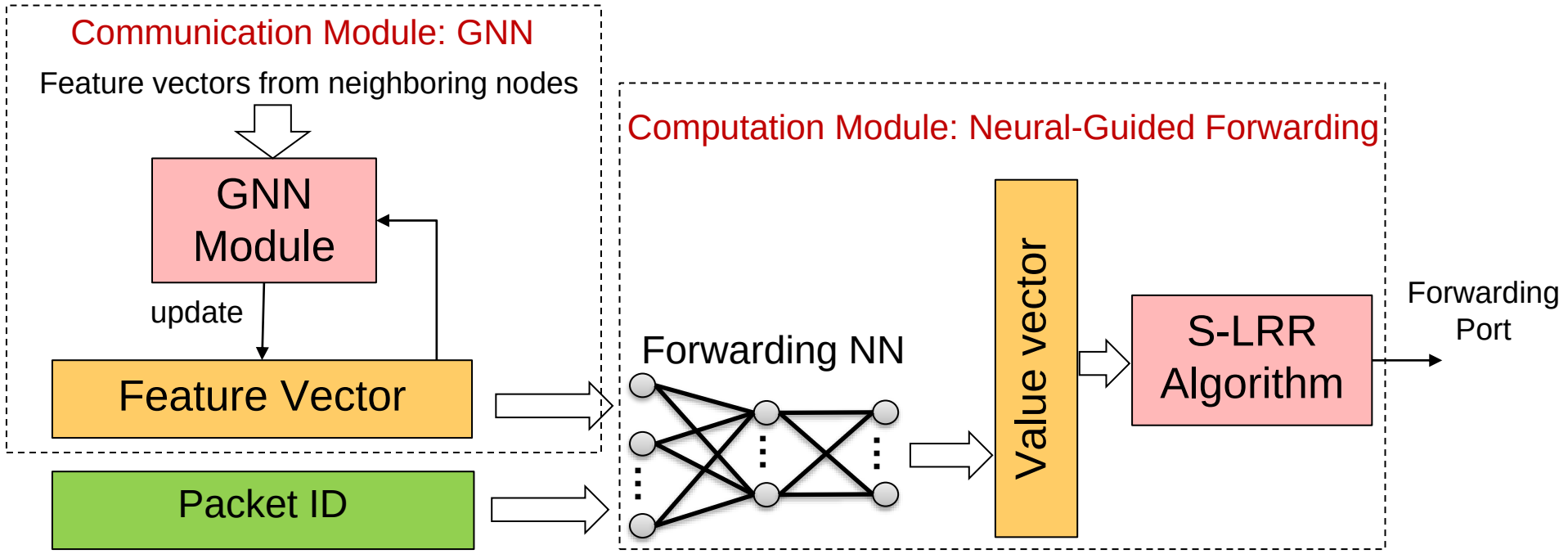
Solution: Neural Guided Routing (NGR)

- Handle the topology changes
  - NGR uses Graph Neural Network (GNN) to handle topology changes
  - When topology changes (e.g., link failures),
    - Step 1. Each node informs its local hidden vector to the neighboring nodes
    - Step 2. Each node aggregates the hidden vectors received from neighboring nodes to update its local hidden vector
    - Step 3. Go to step 1 until the local hidden vector does not change (or reach a pre-defined number of steps)
  - In this way, the new topology information is embedded into the node feature vector



Solution: Neural Guided Routing (NGR)

- Putting all the pieces together



Evaluation

- Topology
  - Internet zoo topologies [Knight et al., 2011]
    - Max node count: 9, Max link count: 20
- NGR parameter
  - Forwarding NN:
    - Input: one-hot vectors of node index and the topology feature vector
    - Three hidden layers
    - Output: the node value
  - GNN:
    - Aggregation: GRU with output feature vector of dim 200
- Compared solution
  - Non-learning based:
    - Shortest path routing (SPF)
    - Triangle-constraint-based routing (PEFT)
  - Learning based:
    - GQNN [Geyer et al., 2018]
- Tasks
  - Load balancing
  - Shortest path routing

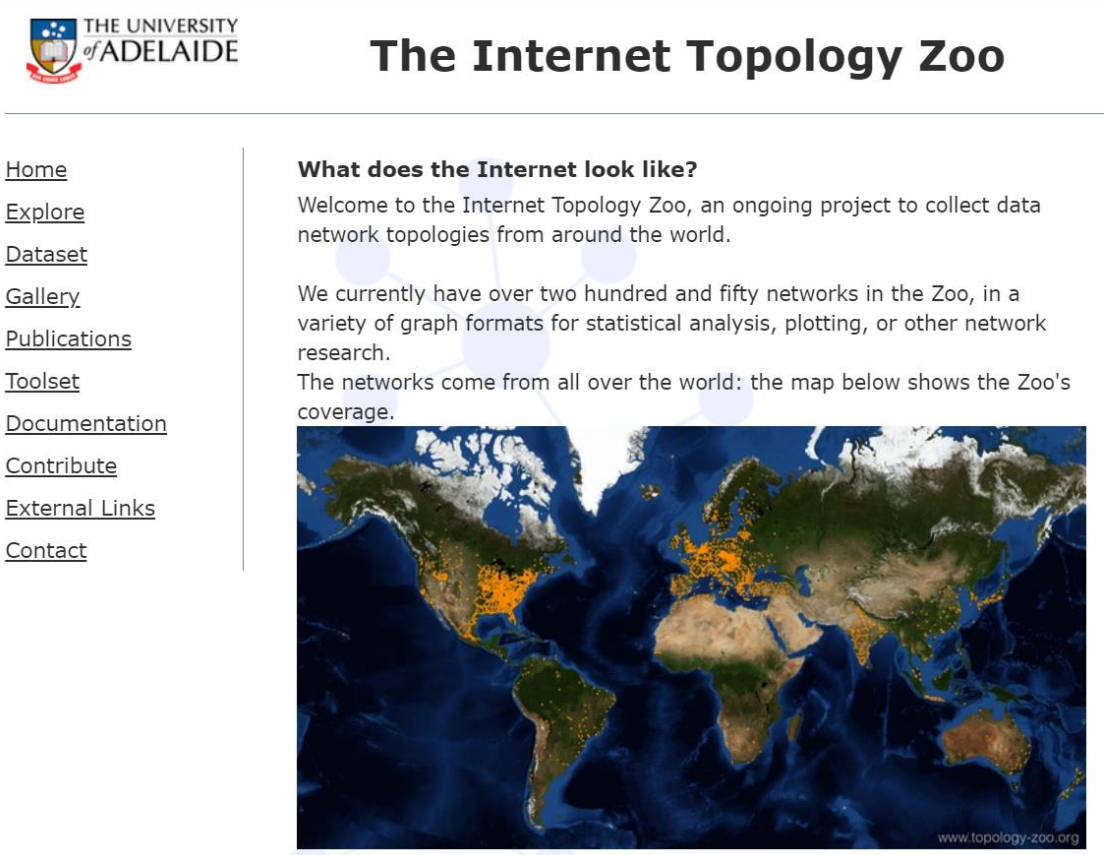
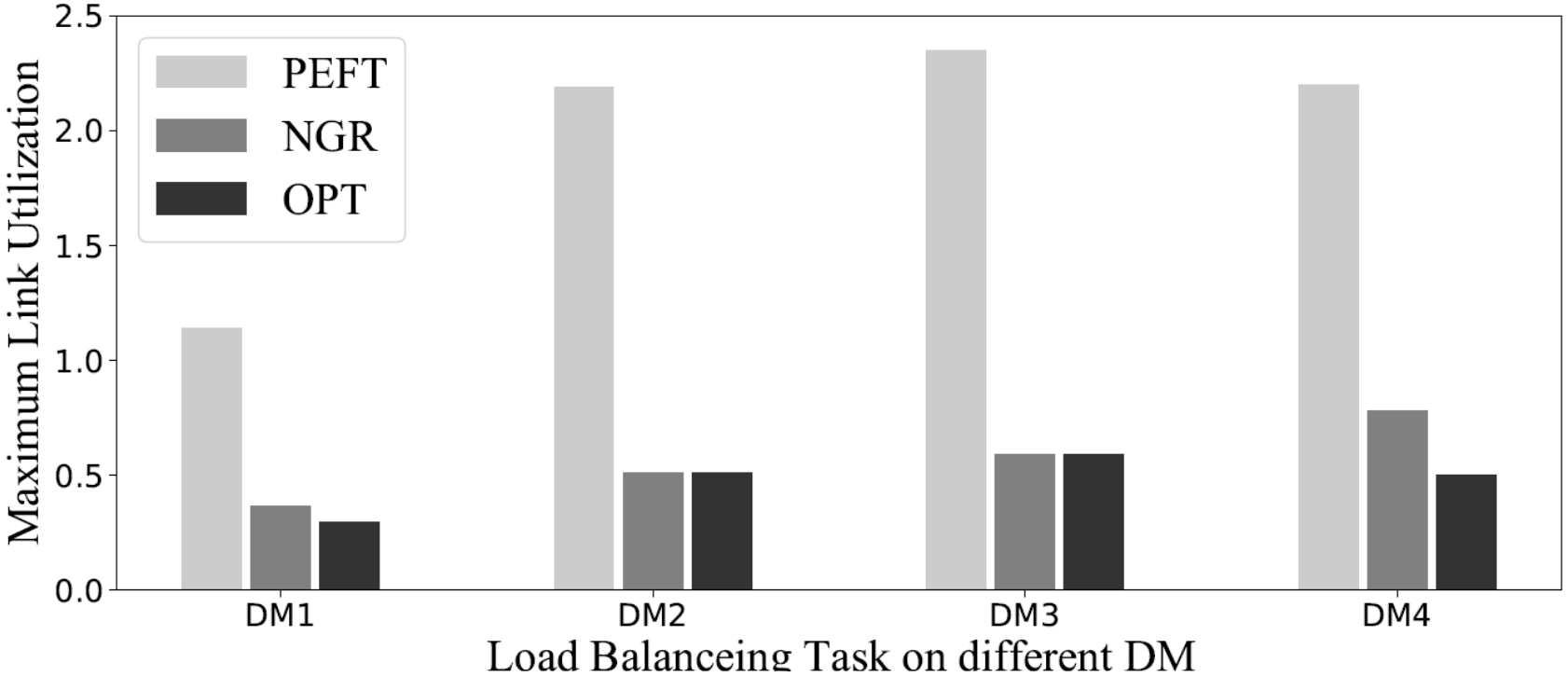


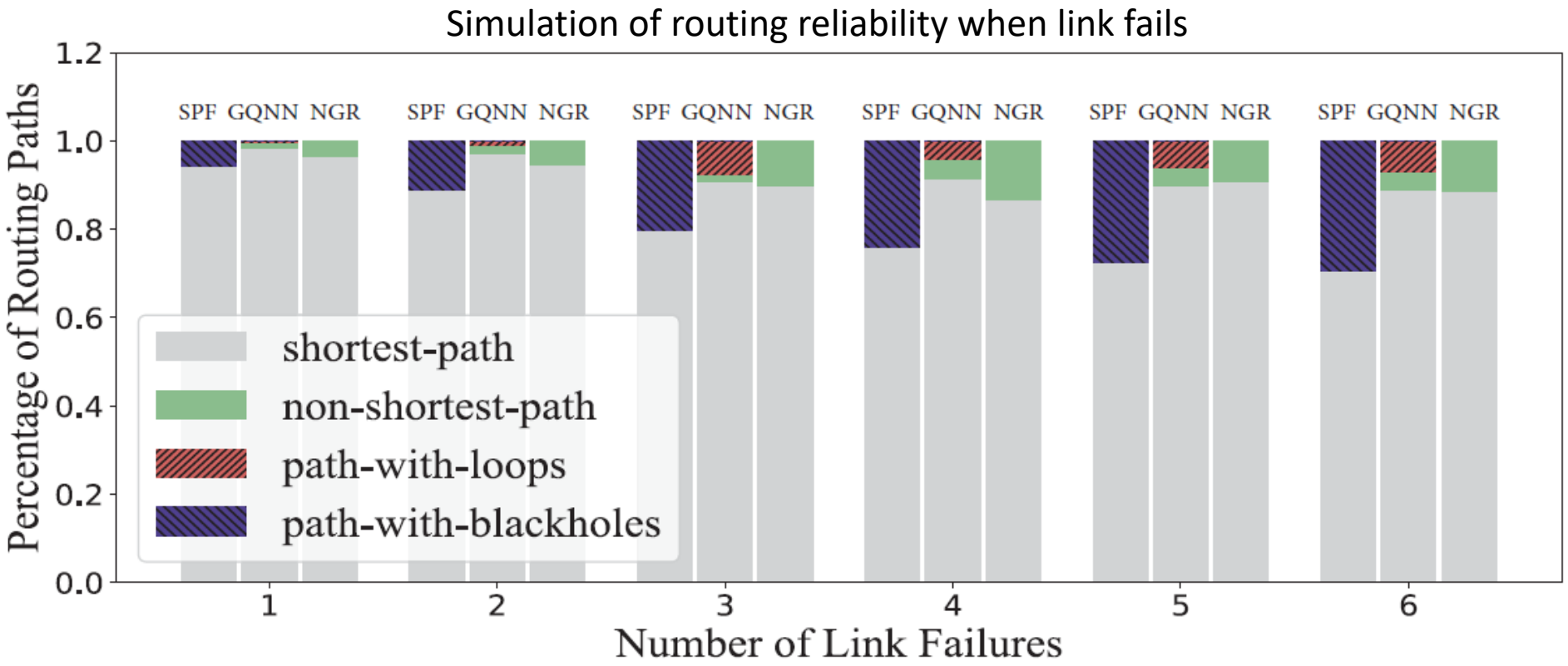
Figure source: http://www.topology-zoo.org/

Evaluation



NGR shows near-optimal performance in different traffic demands

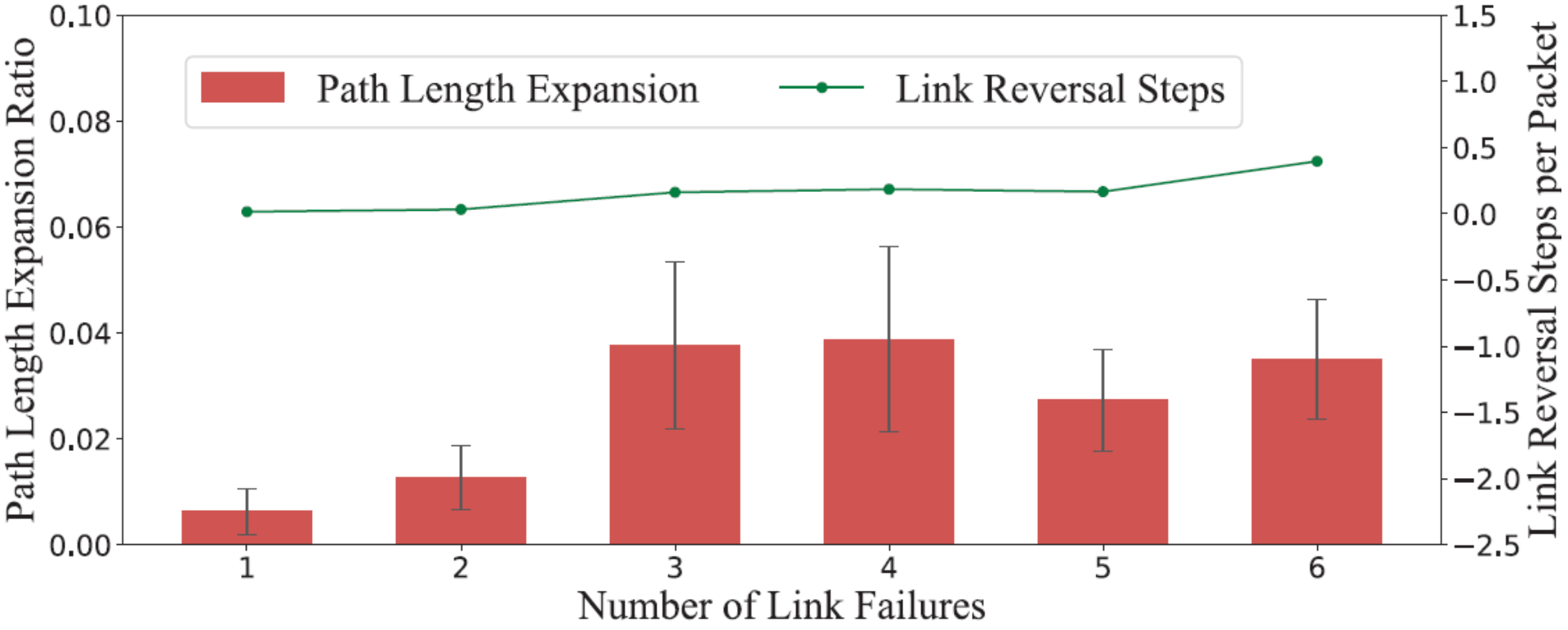
Evaluation



1. SPF has more than twice blackholes that of GQNN and NGR
2. GQNN generates routing loops when more link failures happen due to neural network errors
3. NGR does not generate any loops or blackholes in all the cases, at the cost of slightly longer paths

Evaluation

Simulation of packet-head overheads and path lengths of NGR when link fails



1. NGR achieves a small path length expansion ratio below 6% under all the cases
2. The link reversal overhead in NGR keeps small and increases slowly with more link failures

Conclusion

- NGR: reliable deep-learning-based distributed routing
  - **Reliable**: connectivity guarantee based on the link reversal theory
  - **Flexible**: support flexible customized optimization goals
  - **Optimal**: attain the capacity to achieve the optimality with respect to the given optimization goal
- Future work
  - Reinforcement learning to learn more complex optimization objective (e.g., delay)
  - Scalability and generalization for large-scale network topologies
  - Line-rate neural network inference in network devices

References

- [Swamy et al., 2020] Swamy, T. and Rucker, A. and Shahbaz, M. and Olukotun, K.
  - Taurus: An Intelligent Data Plane
  - *https://arxiv.org/abs/2002.08987*
- [DeepMind, 2018] Jaderberg, M. and Czarnecki, W.M. and Dunning, I. and et al.
  - Human-level performance in first-person multiplayer games with population-based deep reinforcement learning
  - *https://arxiv.org/abs/1807.01281*
- [Geyer et al., 2018] Geyer, F. and Carle, G.
  - Learning and Generating Distributed Routing Protocols Using Graph-Based Deep Learning
  - *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. 2018
- [Liu et al. 2013] Liu, J., Panda, A., Singla, A., Godfrey, B., Schapira, M., & Shenker, S.
  - Ensuring Connectivity via Data Plane Mechanisms
  - *NSDI 2013*
- [Knight et al., 2011] Knight, S. and Nguyen, H.X. and Falkner, N. and Bowden, R. and Roughan, M.
  - The Internet Topology Zoo
  - *IEEE J. Sel. Areas Commun., 29(9):1765–1775.*

Thank You !