Unveiling the potential of Graph Neural Networks

Prof. Albert Cabellos (acabello@ac.upc.edu)

bnn.upc.edu

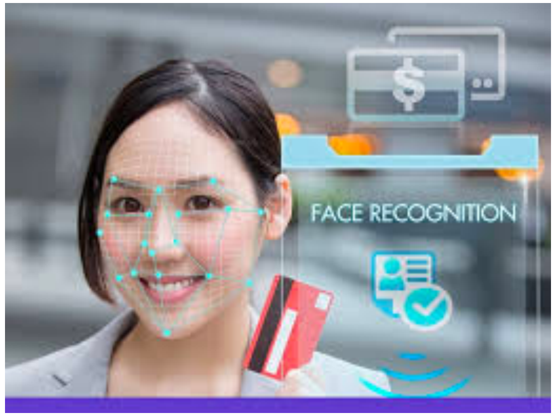
Universitat Politècnica de Catalunya

ESNOG/GORE 24, CSUC

October 2019

ML Applied to Networking: What can we learn from Computer Vision?

Computer Vision - A successful application of AI



Facial Recognition

Self-driving Cars

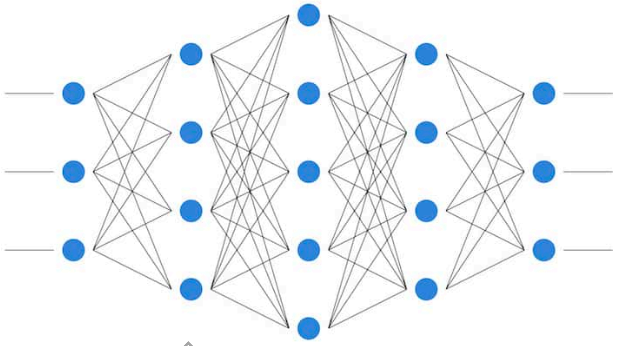
- Computer Vision automates tasks that the human vision can do.
- Excellent research, innovation that lead to a breakthrough in products and services.
- We aim to apply ML to Networks

What can we learn from computer vision?

How does Computer Vision work?

2

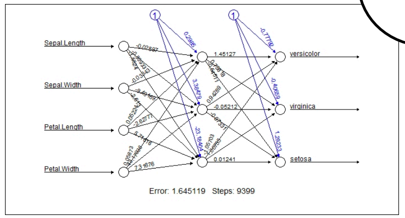
Convolutional Neural Network



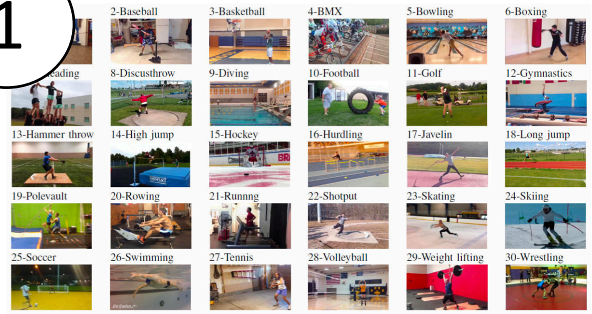
Trained Model

3

Training



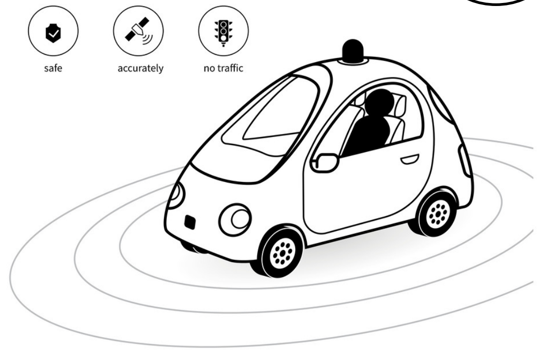
1



Labeled Dataset
(e.g, ImageNet)

4

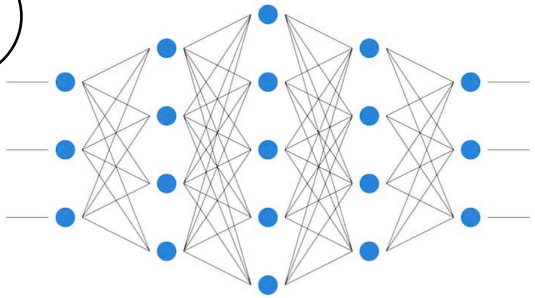
SELF-DRIVING CAR



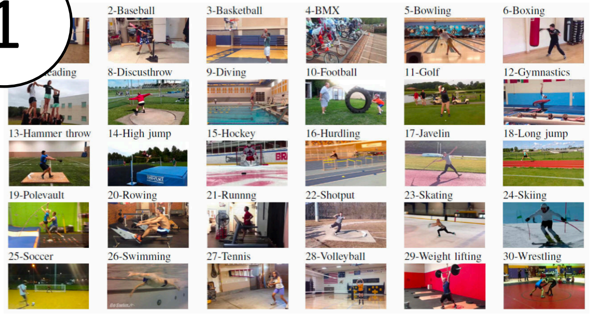
ImageNet and AlexNet

2

Convolutional Neural Network



1



Labeled Dataset
(e.g, ImageNet)

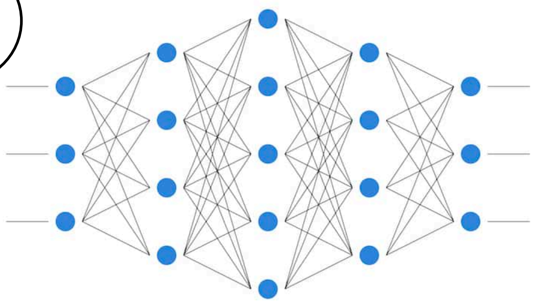
- AI research, education or innovation requires:

1. A dataset
2. A specific type of neural network that can learn the structure of the information (e.g, images, voice, graphs, etc)

ImageNet and AlexNet

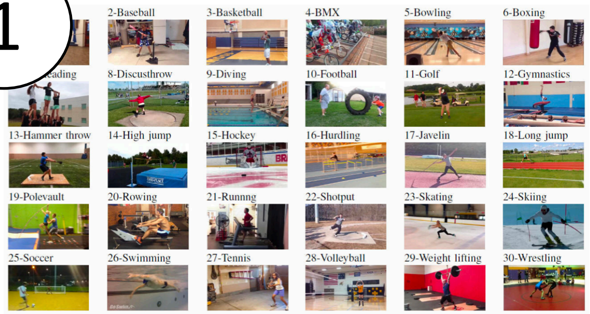
2

Convolutional Neural Network



A specific type of neural network: Convolutional Neural Network

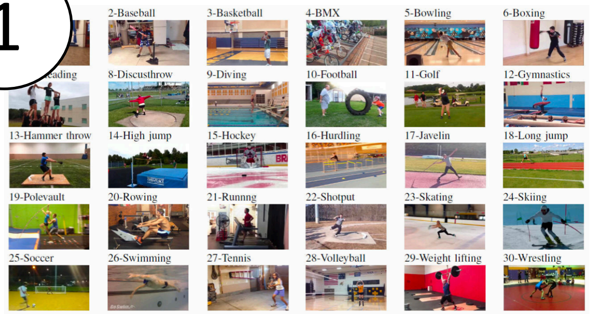
1



Labeled Dataset
(e.g, ImageNet)

A Dataset: A repository of labeled images

1



ImageNet

Labeled Dataset
(e.g, ImageNet)

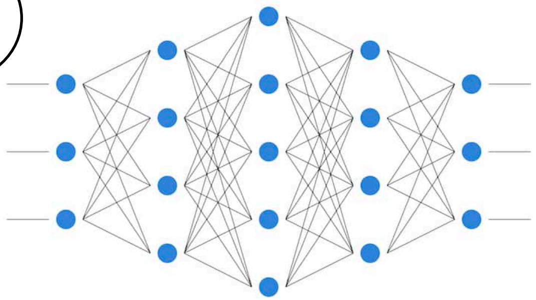
IMAGENET

http://www.image-net.org/

- A public repository of 14M labeled images
- A fundamental tool for **education** and **research**
- Required for **benchmarking**
- Organize the Large Scale Visual Recognition Challenge
  - Competition among top-players in the field (Cambridge, Microsoft, etc)

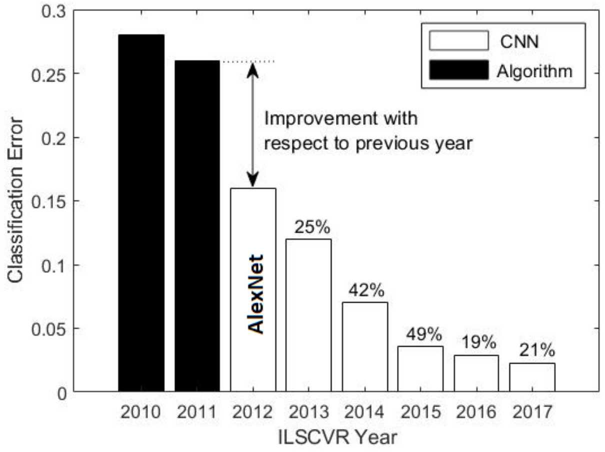
2

Convolutional Neural Network



AlexNet

- A pioneering Convolutional Neural Network
- A breakthrough in Computer Vision
- 40k citations (most cited papers in Computer Sciences)



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

1 Introduction

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on the order of tens of thousands of images (e.g., NORB [16], Caltech-101/256 [8, 9], and CIFAR-10/100 [12]). Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current-best error rate on the MNIST digit-recognition task (<0.3%) approaches human performance [4]. But objects in realistic settings exhibit considerable variability, so to learn to recognize them it is necessary to use much larger training sets. And indeed, the shortcomings of small image datasets have been widely recognized (e.g., Pinto et al. [21]), but it has only recently become possible to collect labeled datasets with millions of images. The new larger datasets include LabelMe [23], which consists of hundreds of thousands of fully-segmented images, and ImageNet [6], which consists of over 15 million labeled high-resolution images in over 22,000 categories.

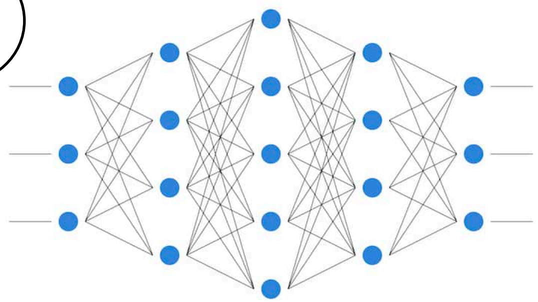
To learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don't have. Convolutional neural networks (CNNs) constitute one such class of models [16, 11, 13, 18, 15, 22, 26]. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

ImageNet and AlexNet

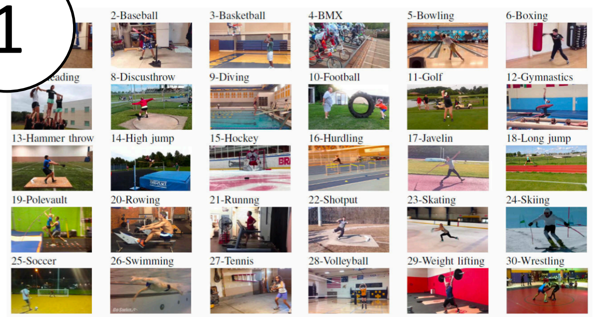
2

Convolutional Neural Network



AlexNet

1



Labeled Dataset
(e.g, ImageNet)

ImageNet

Barcelona Neural Networking Center

Barcelona
Neural
Networking
Center

- Our goal is to become the **enablers** in the application of ML to networking
- How?

Build **AlexNet** and **ImageNet** of Computer Networks

- What is our competitive advantage?

Graph Neural Networks

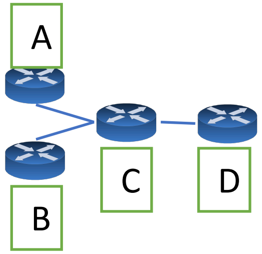
What are
Graph Neural Networks?

ML applied to Networking

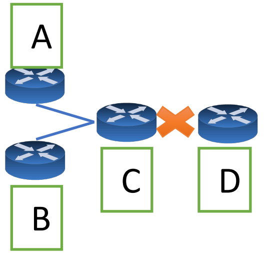
- So far we have **<u>failed</u>** to learn Computer Networks (e.g)
  - Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017, November). Learning to route. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks* (pp. 185-191). ACM.
  - Chen, X., Guo, J., Zhu, Z., Proietti, R., Castro, A., & Yoo, S. J. B. (2018, March). Deep-RMSA: A Deep-Reinforcement-Learning Routing, Modulation and Spectrum Assignment Agent for Elastic Optical Networks. In *2018 Optical Fiber Communications Conference and Exposition (OFC)* (pp. 1-3). IEEE.
  - Poor performance, in some cases worse than simple well-known heuristics
  - Ad-hoc solutions tailored to specific problems, in some cases transforming the problem to prevent learning graphs

The main reason for this is that standard Neural Networks **<u>are not suited</u>** to learn information structured as a graph

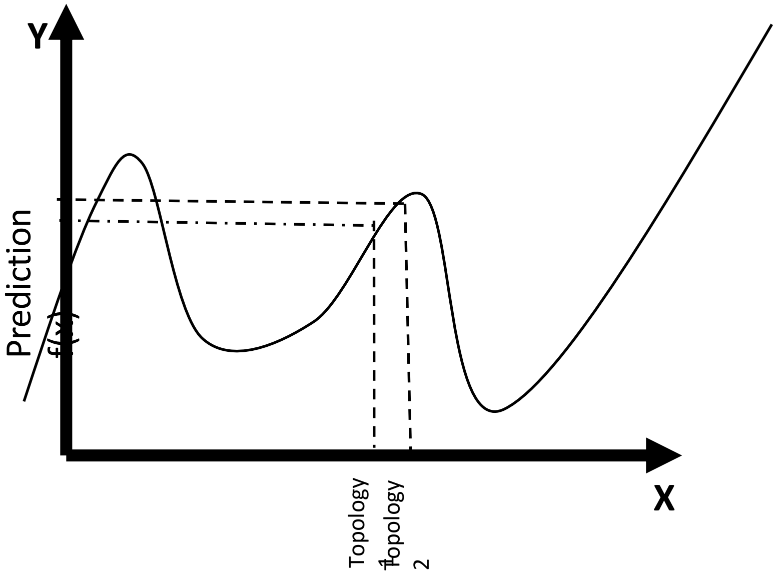
Graph Neural Networks



	A	B	C	D
A	1	0	1	0
B	0	1	1	0
C	1	1	1	1
D	0	0	1	1



	A	B	C	D
A	1	0	1	0
B	0	1	1	0
C	1	1	1	0
D	0	0	0	1



- **Networks are graphs:**
  - Routing
  - Topology
  - Etc.
- Learning Graphs with Fully Connected Neural Networks is very complex
- Academics have repeatedly failed to achieve this

Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. *In ACM SOSR 2019*

https://arxiv.org/pdf/1901.08113.pdf

Graph Neural Networks (GNN)

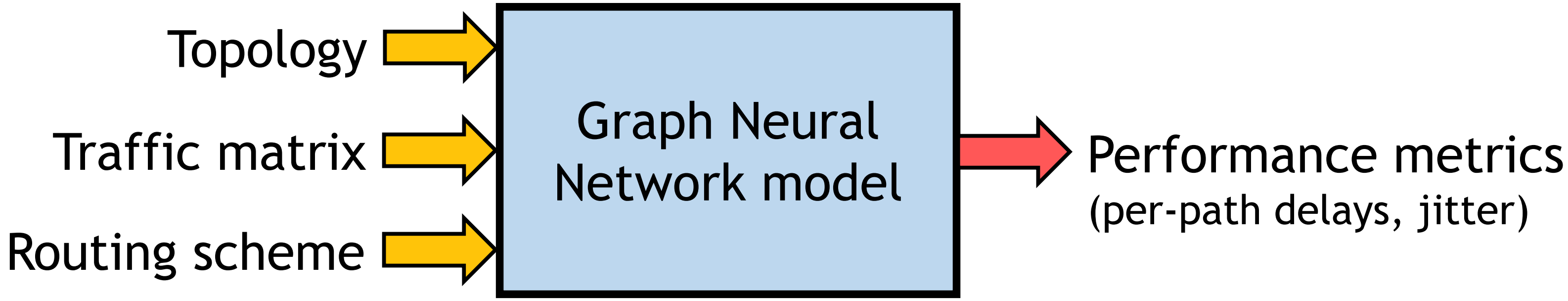
- GNN have been recently proposed by DeepMind *et al.* to learn and model information structured as a **graph**

Component	Entities	Relations	Rel. inductive bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

- Each application has developed their own NN architectures
  - Fully Connected = Units → General application (non-linear regression)
  - CNN = Grid elements → Images
  - RNN = Sequences → Text processing, Time-Series
  - GNN = Nodes + Edges → Networks

Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261*(2018).

RouteNet: The first GNN for Computer Networks

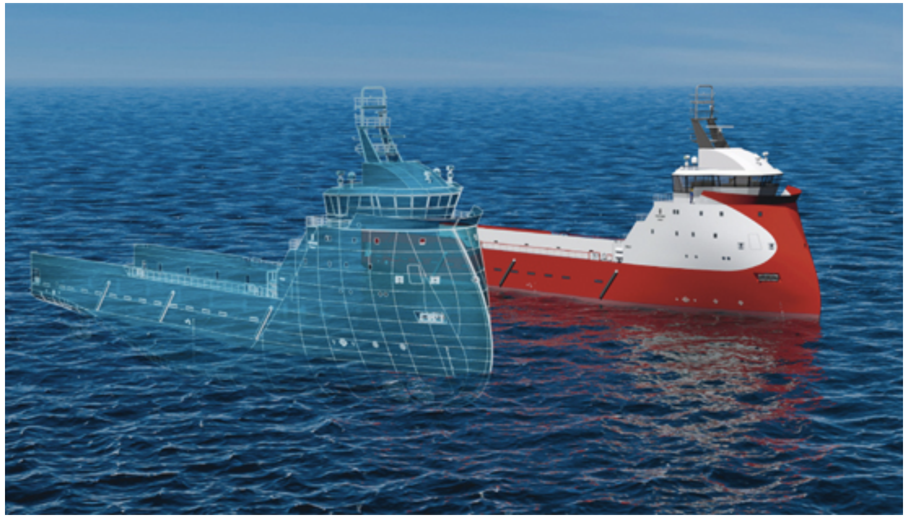


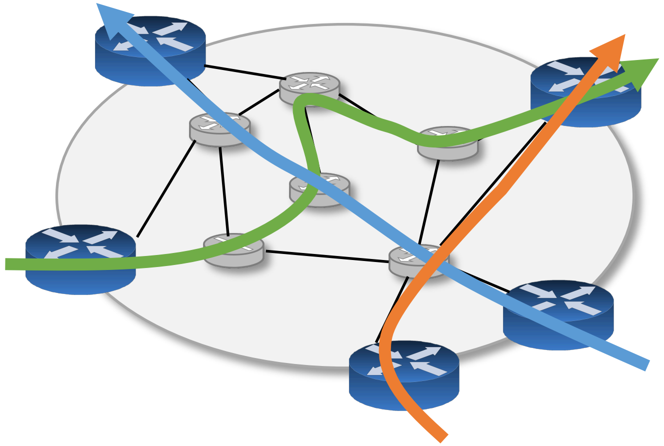
- RouteNet is the first Graph Neural Network for Computer Networks
- It learns the relationship between topology, traffic, routing and the resulting performance of the network
- Generalizes to **unseen** topologies, routings and traffics

Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. *In ACM SOSR 2019*
https://arxiv.org/pdf/1901.08113.pdf

How does RouteNet work?

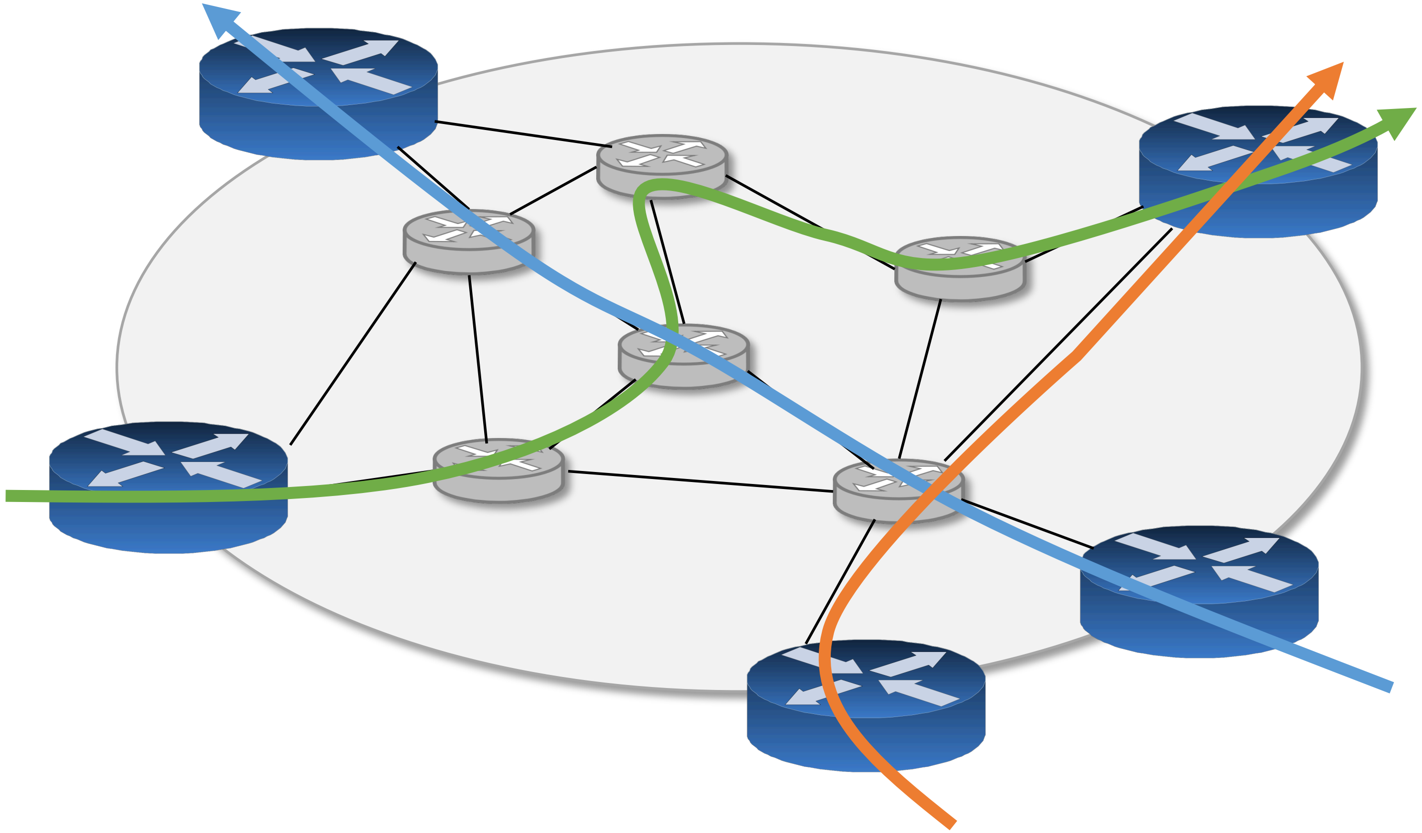
RouteNet: A Digital Network Twin



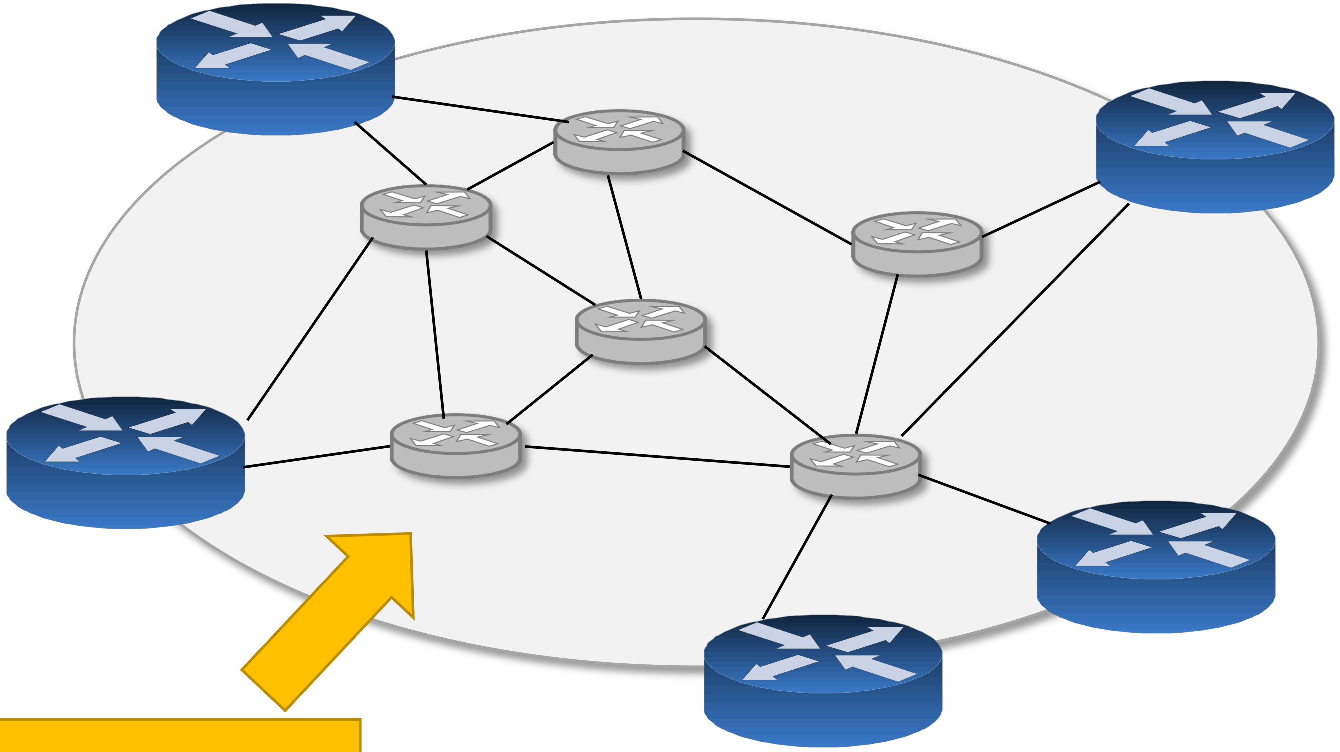


- Digital Twin is a digital representation of the network
- RouteNet is a Digital Twin of the networking infrastructure
- Useful to:
  - What-if scenarios:
    - predict losses for a particular traffic load
    - What happens if a link fails?
  - Optimal configurations:
    - Best routing configuration to load-balance utilization
  - Network Planning

RouteNet: A Digital Twin of the Network

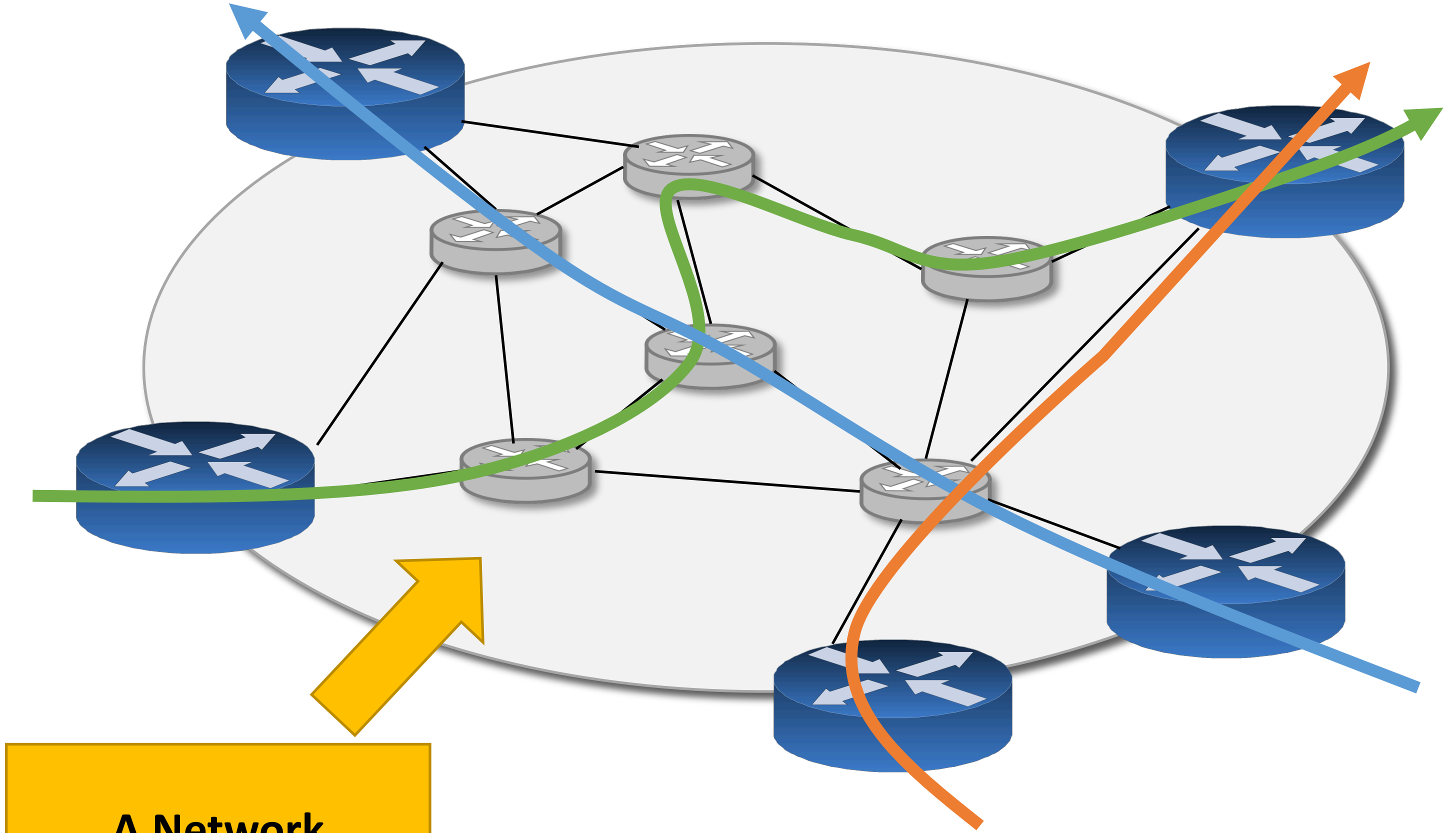


RouteNet: A Digital Twin of the Network



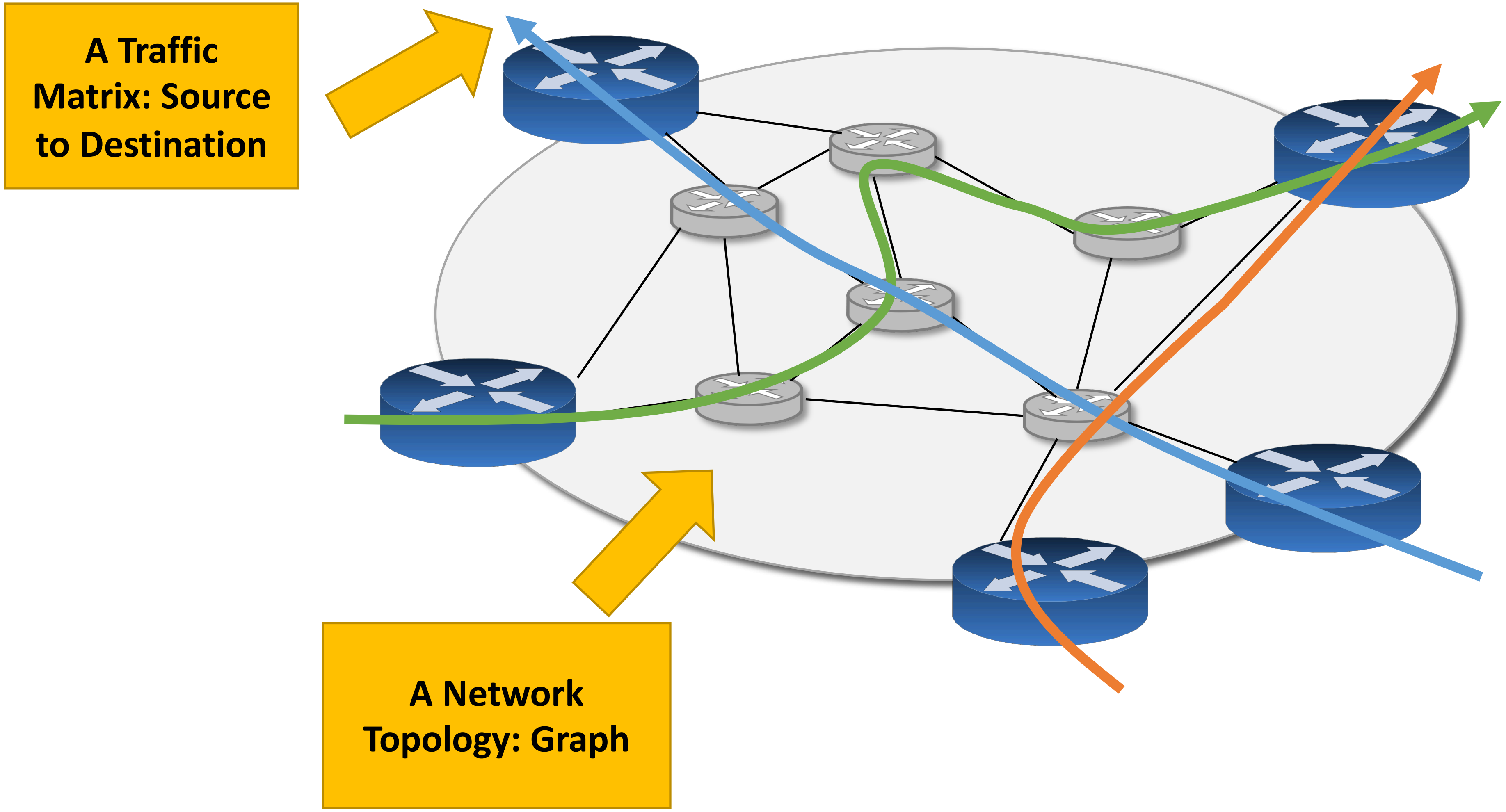
A Network
Topology: Graph

RouteNet: A Digital Twin of the Network

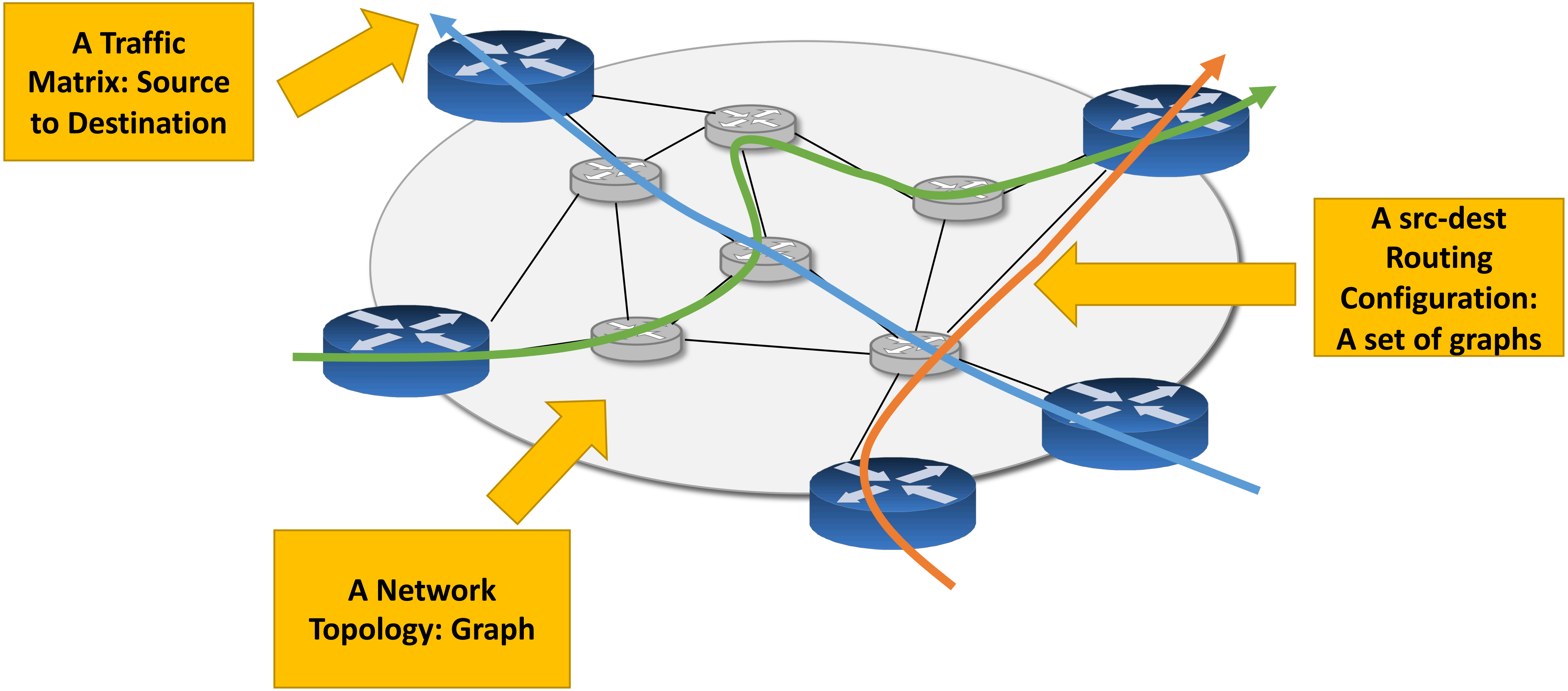


A Network
Topology: Graph

RouteNet: A Digital Twin of the Network

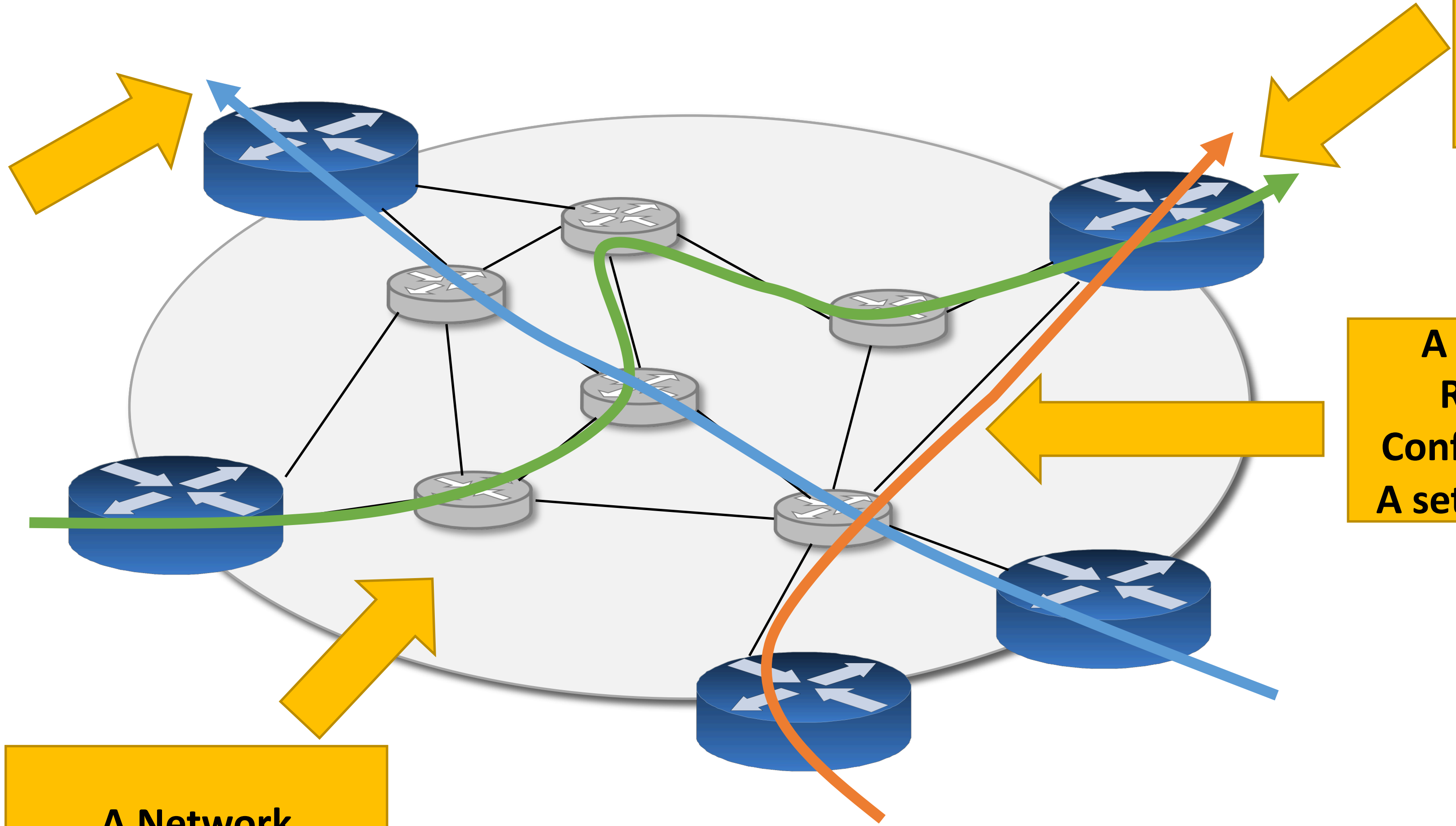


RouteNet: A Digital Twin of the Network



RouteNet: A Digital Twin of the Network

A Traffic Matrix: Source to Destination

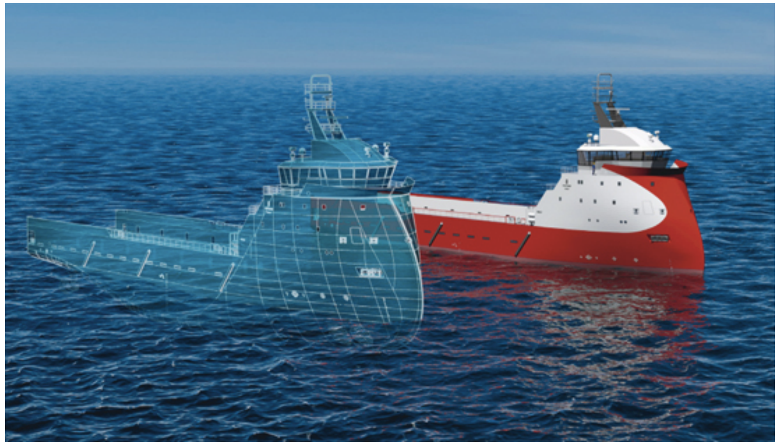


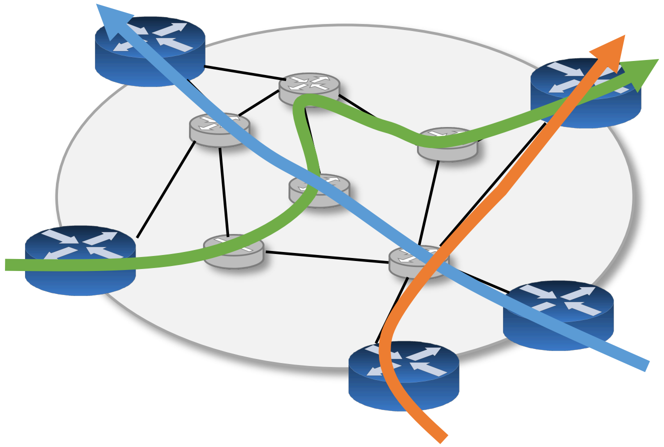
Performance Metrics: Per src-dest delay, jitter and drops

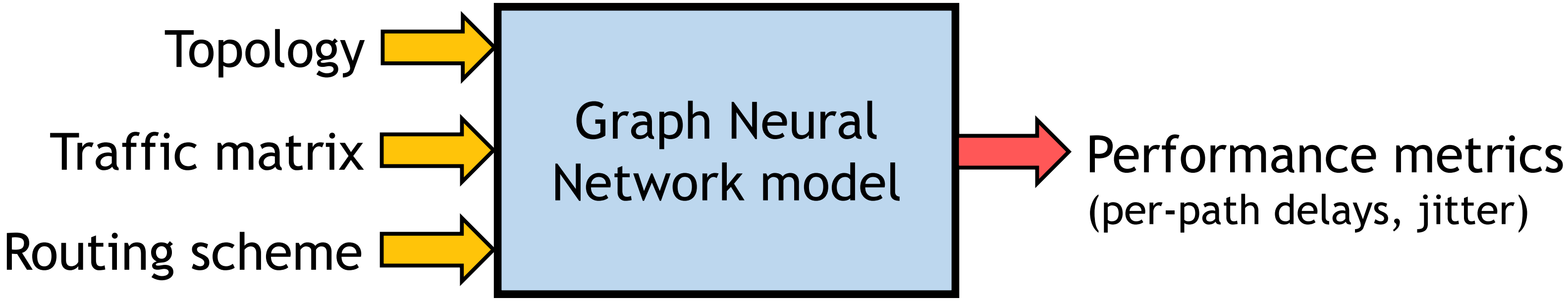
A src-dest Routing Configuration: A set of graphs

A Network Topology: Graph

RouteNet: The first GNN for Computer Networks

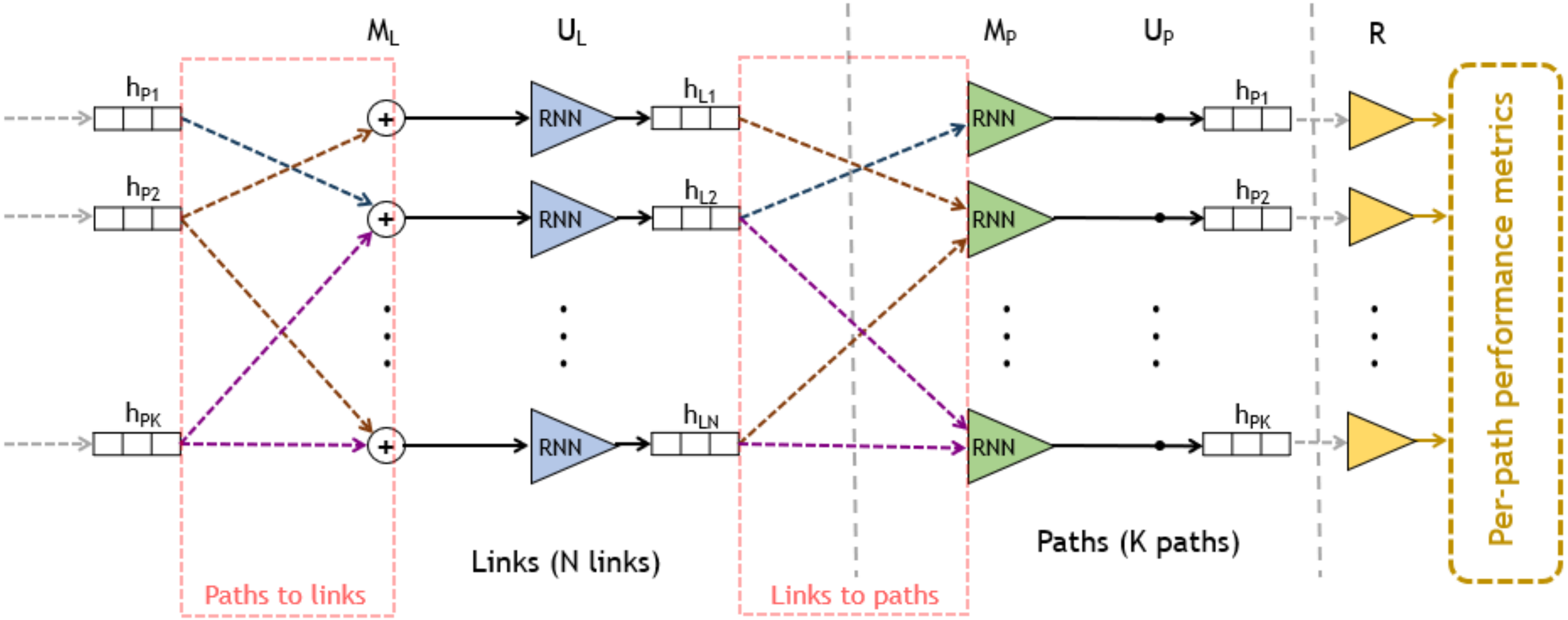






Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. *In ACM SOSR 2019*
https://arxiv.org/pdf/1901.08113.pdf

RouteNet Architecture

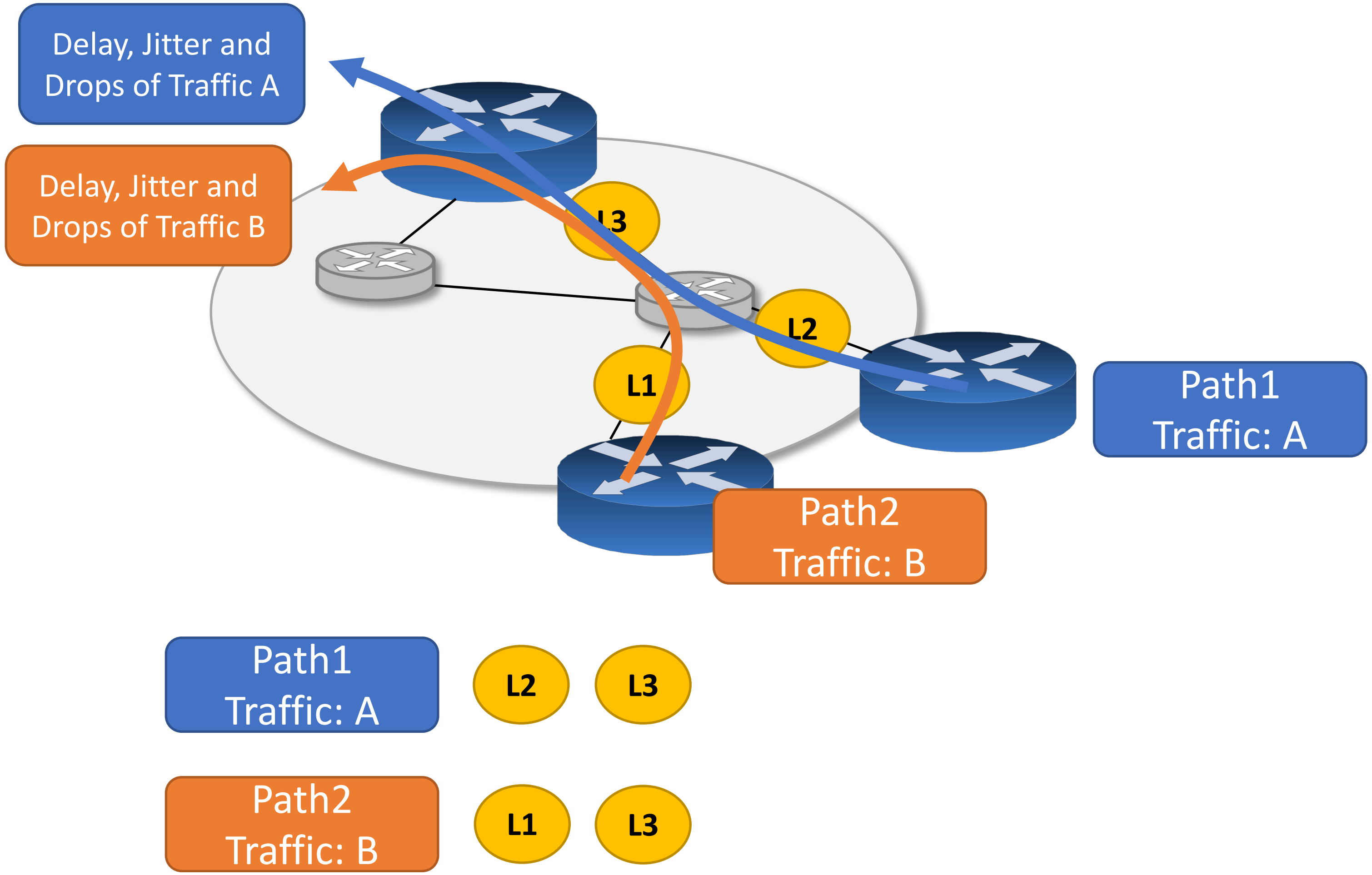


Input: $\mathbf{x}_p, \mathbf{x}_l, \mathcal{R}$
Output: $\mathbf{h}_p^T, \mathbf{h}_l^T, \hat{\mathbf{y}}_p$
foreach $p \in \mathcal{R}$ **do**
  $\mathbf{h}_p^0 \leftarrow [\mathbf{x}_p, 0 \ldots, 0]$
end
foreach $l \in \mathcal{N}$ **do**
  $\mathbf{h}_l^0 \leftarrow [\mathbf{x}_l, 0 \ldots, 0]$
end
for $t = 1$ *to* T **do**
  foreach $p \in \mathcal{R}$ **do**
    foreach $l \in p$ **do**
      $\mathbf{h}_p^t \leftarrow RNN_t(\mathbf{h}_p^t, \mathbf{h}_l^t)$
      $\tilde{\mathbf{m}}_{p,l}^{t+1} \leftarrow \mathbf{h}_p^t$
    end
    $\mathbf{h}_p^{t+1} \leftarrow \mathbf{h}_p^t$
  end
  foreach $l \in \mathcal{N}$ **do**
    $\mathbf{m}_l^{t+1} \leftarrow \sum_{p: l \in p} \tilde{\mathbf{m}}_{p,l}^{t+1}$
    $\mathbf{h}_l^{t+1} \leftarrow U_t\left(\mathbf{h}_l^t, \mathbf{m}_l^{t+1}\right)$
  end
end
$\hat{\mathbf{y}}_p \leftarrow F_p(\mathbf{h}_p)$

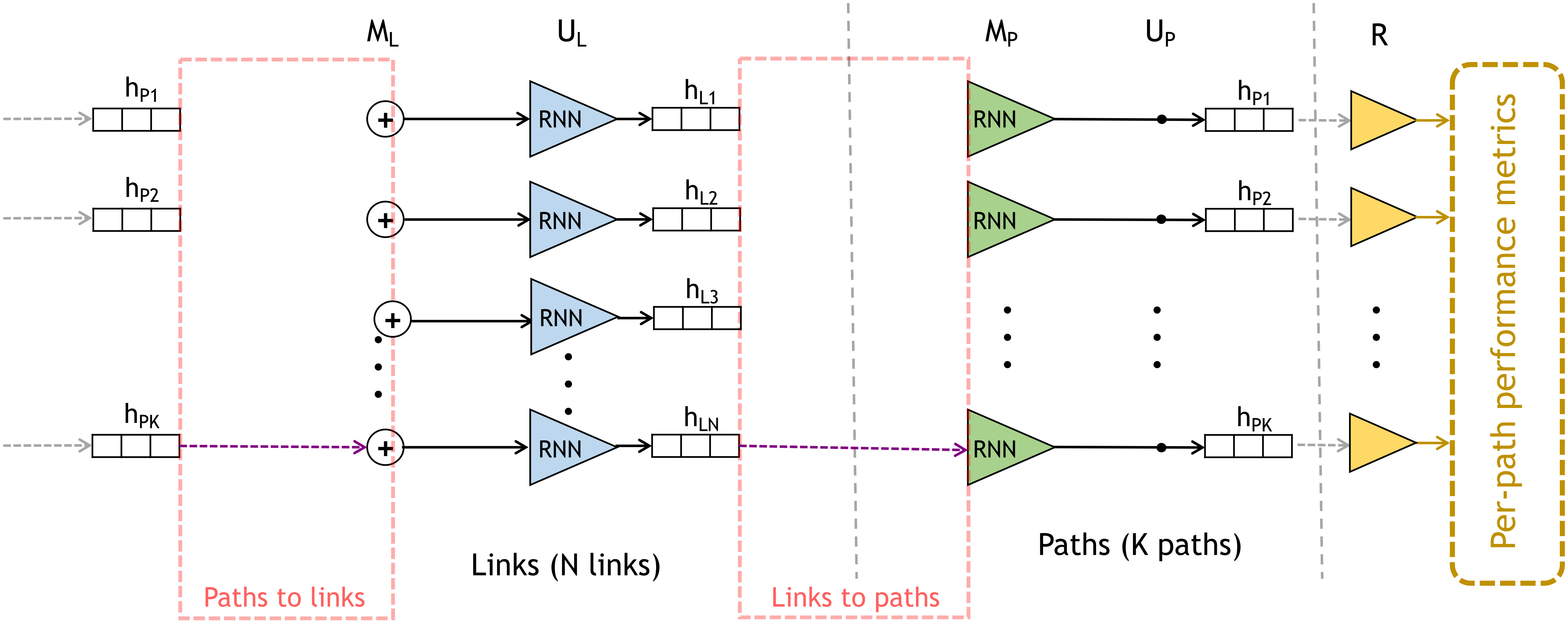
- RoueNet models the relationship between links and paths
  - State of a links depends on the paths that traverse that link
  - State of a paths depends on the links of that path
- This is a circular dependency

$$\mathbf{h}_{l_i} = f(\mathbf{h}_{p_1}, \ldots, \mathbf{h}_{p_j}), \quad l_i \in p_k, k = 1, \ldots, j$$
$$\mathbf{h}_{p_k} = g(\mathbf{h}_{l_{k(0)}}, \ldots, \mathbf{h}_{l_{k(|p_k|)}})$$

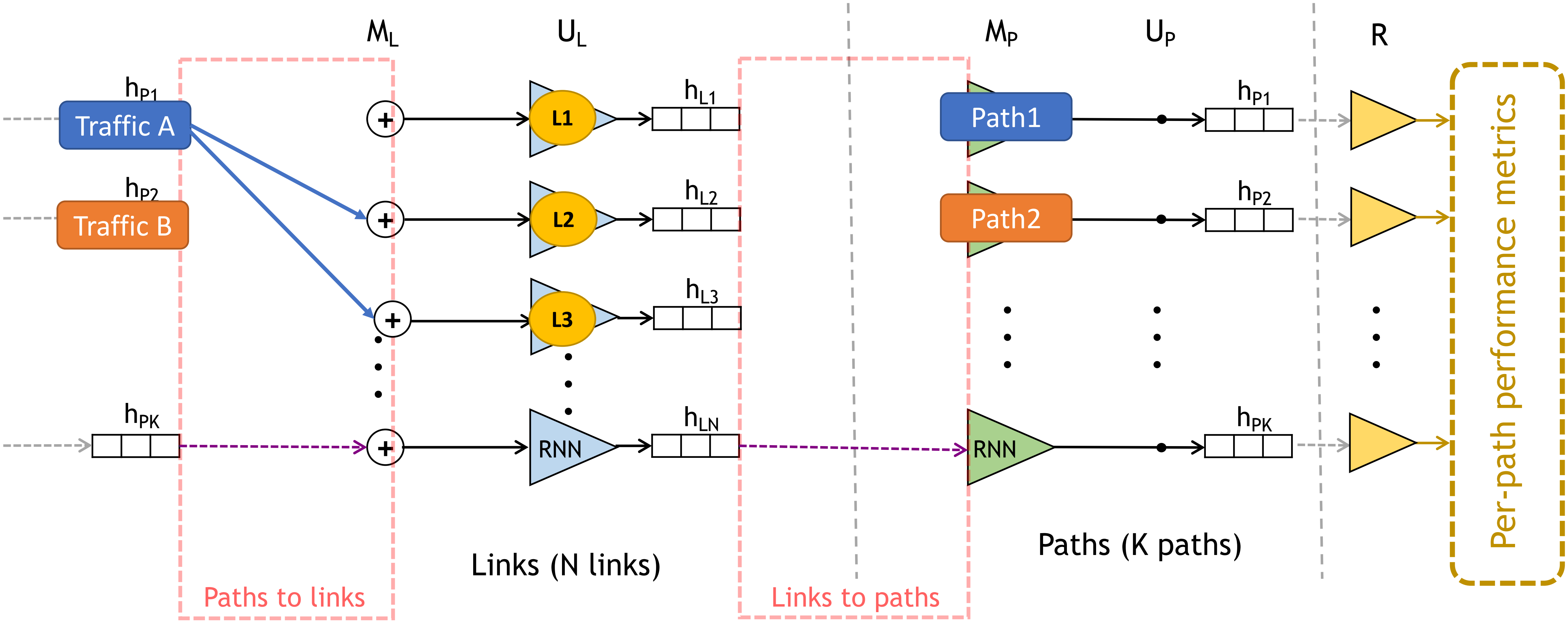
RouteNet: A working example



RouteNet Architecture

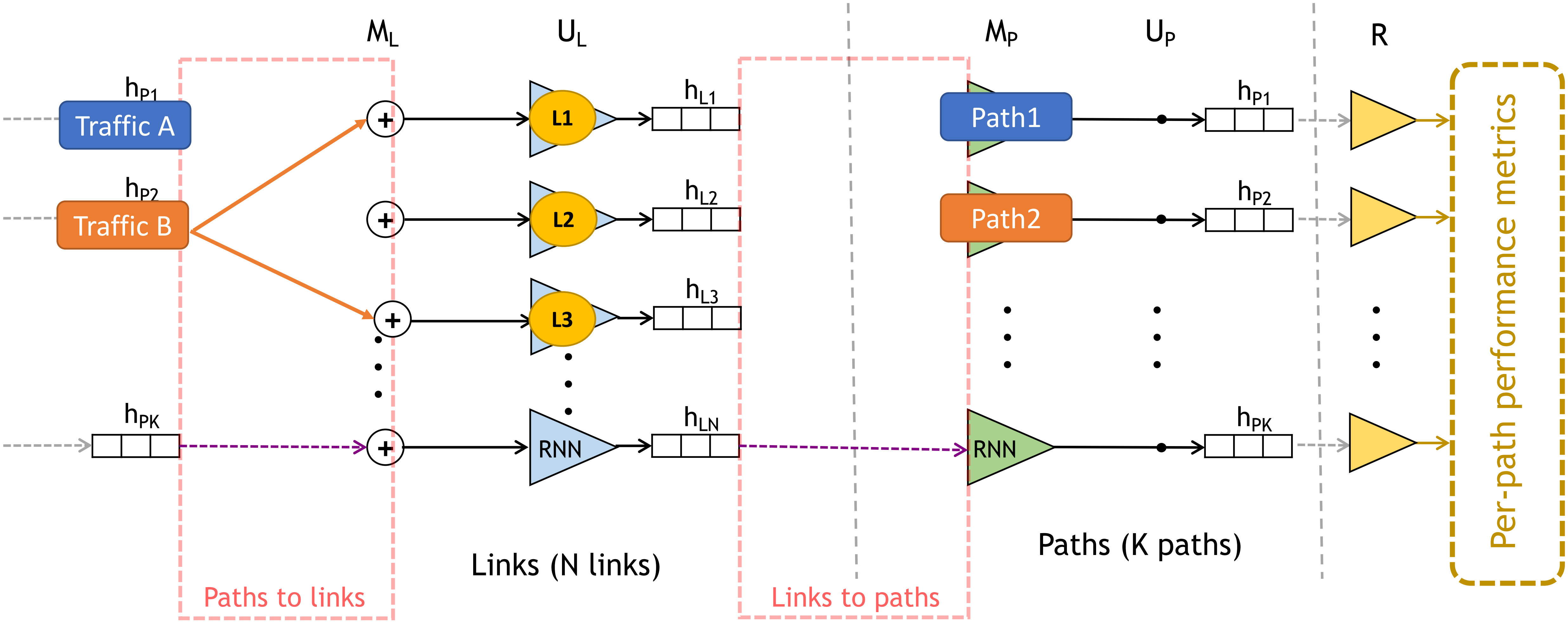


RouteNet Architecture



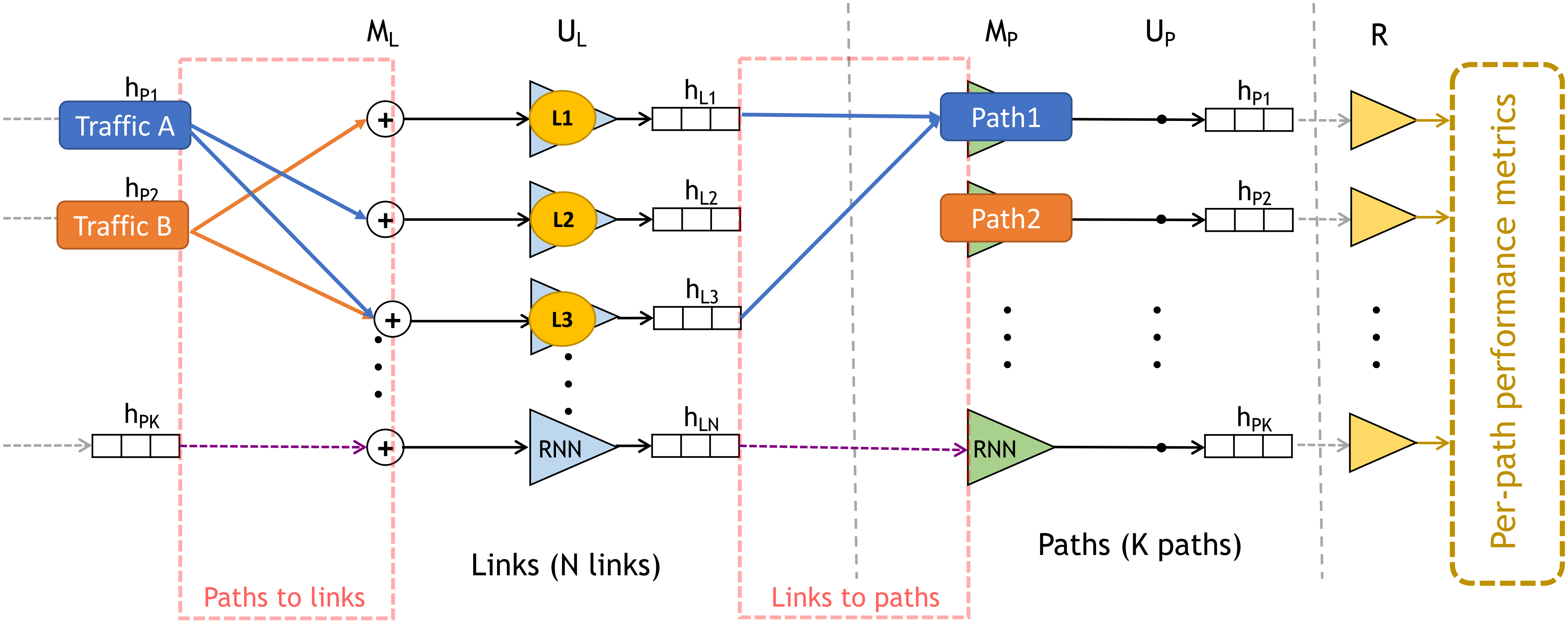
State of a links depends on the paths that traverse that link

RouteNet Architecture



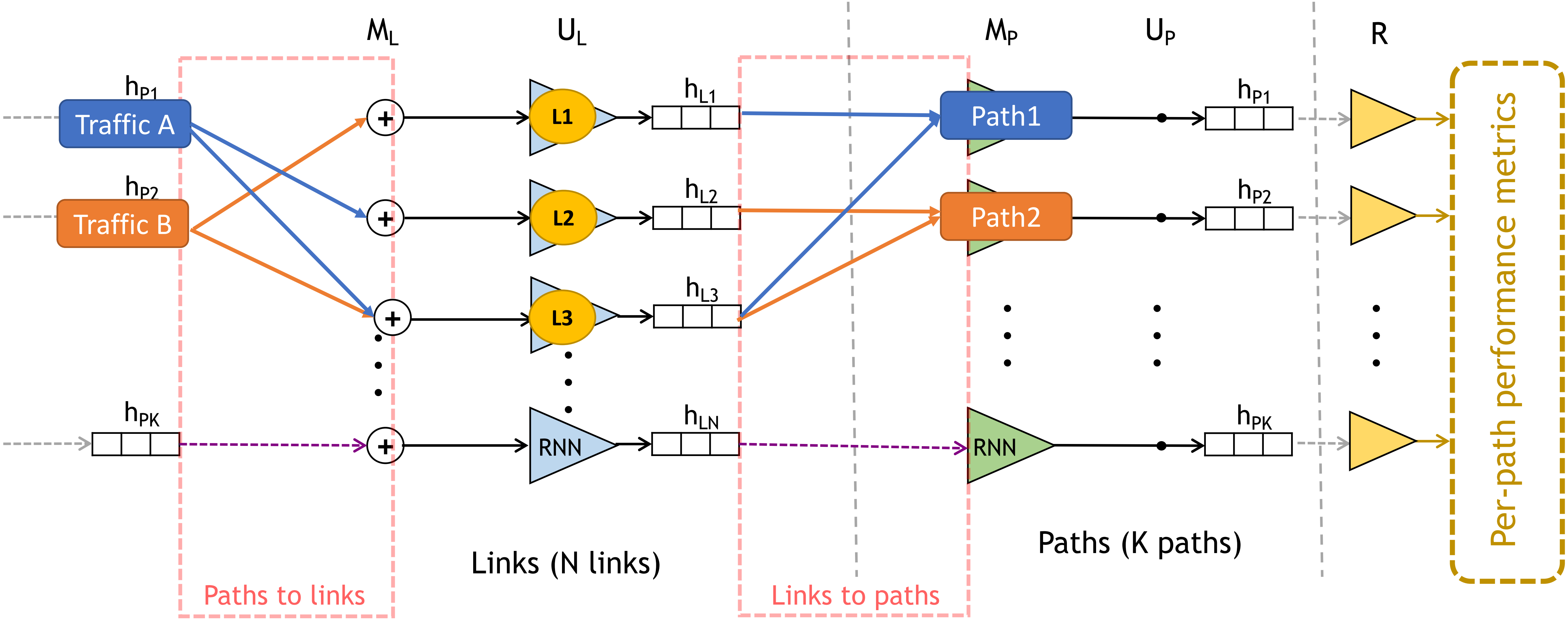
State of a links depends on the paths that traverse that link

RouteNet Architecture



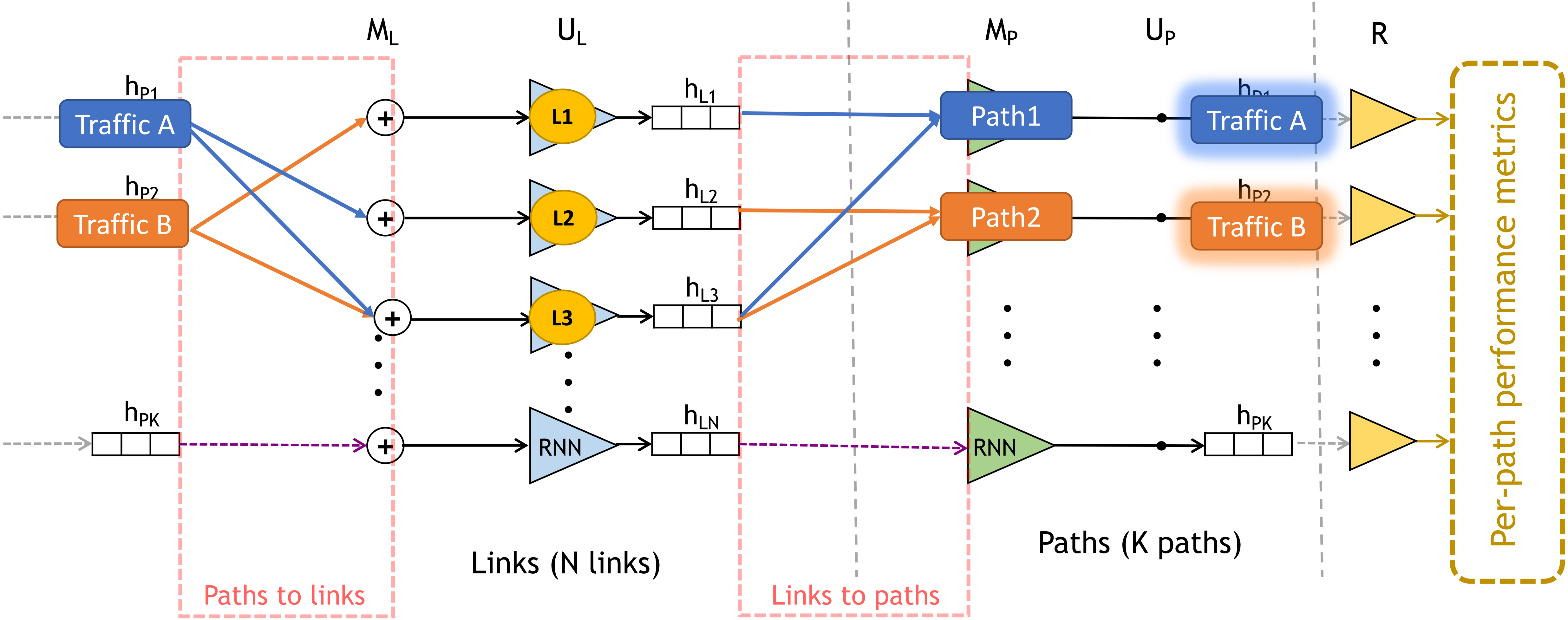
State of a paths depends on the links of that path

RouteNet Architecture

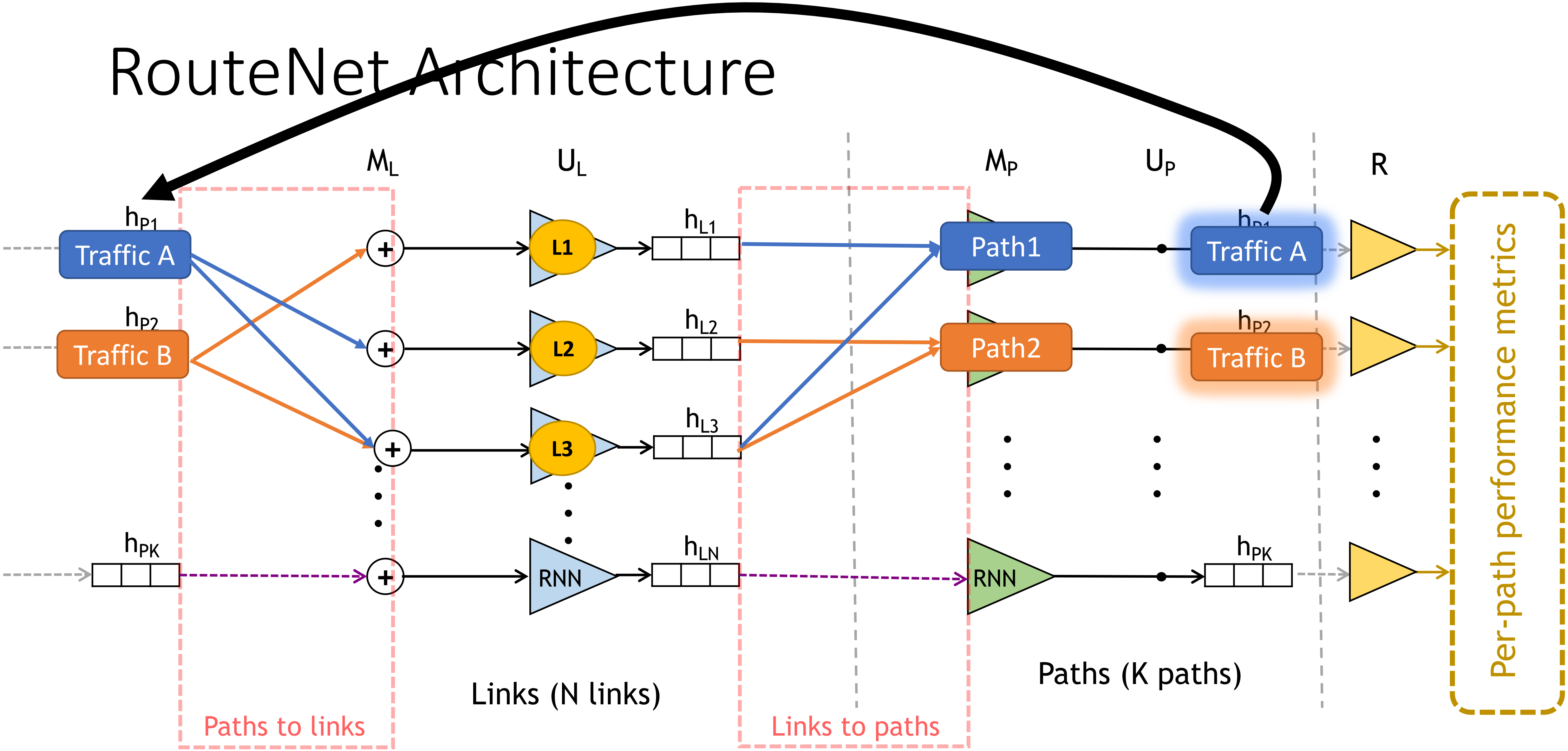


State of a paths depends on the links of that path

RouteNet Architecture

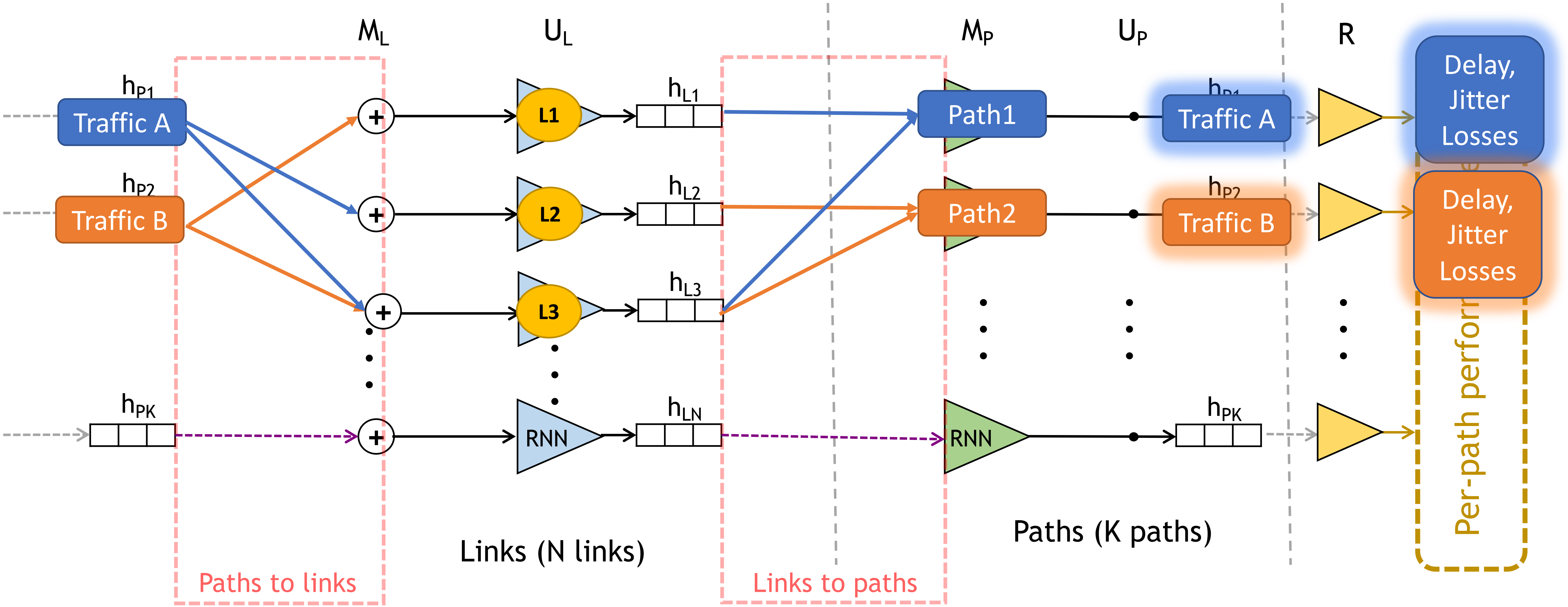


RouteNet Architecture

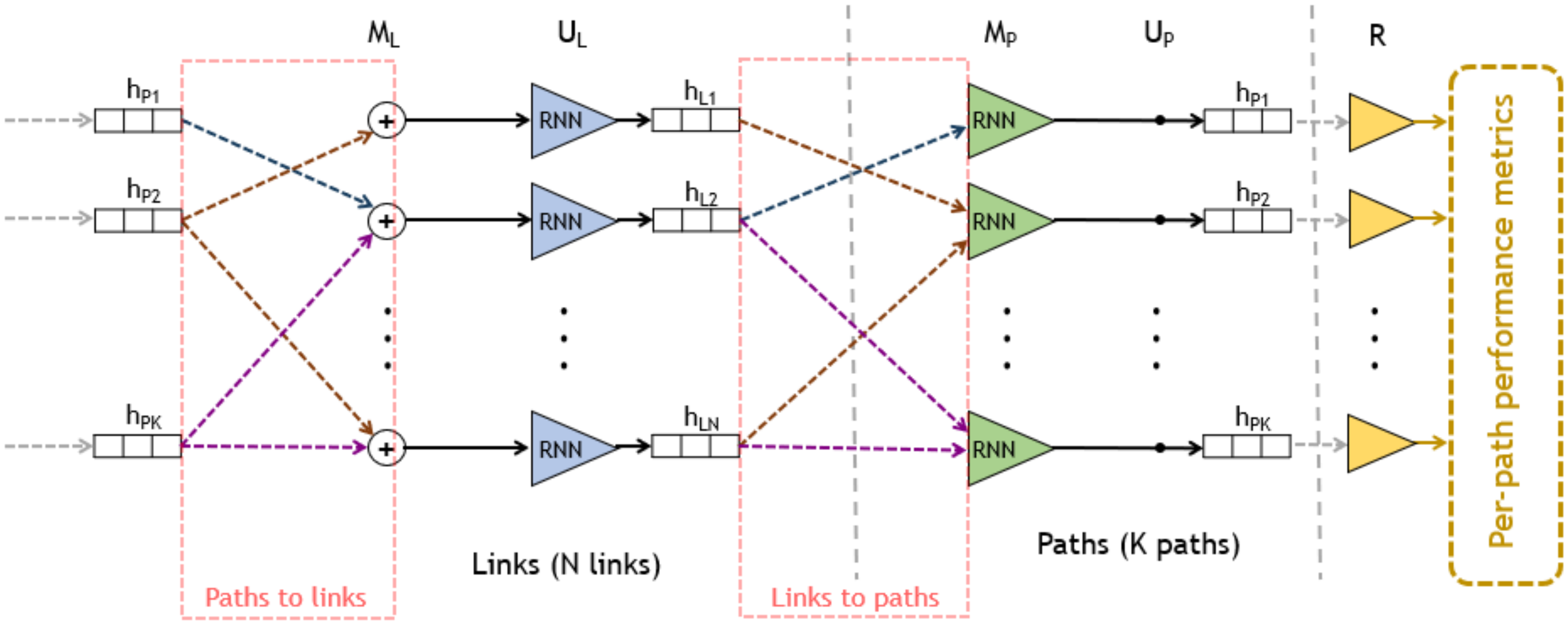


This is a circular dependency

RouteNet Architecture



RouteNet Architecture



Input: $\mathbf{x}_p, \mathbf{x}_l, \mathcal{R}$

Output: $\mathbf{h}_p^T, \mathbf{h}_l^T, \hat{\mathbf{y}}_p$

foreach $p \in \mathcal{R}$ **do**
  $\mathbf{h}_p^0 \leftarrow [\mathbf{x}_p, 0 \ldots, 0]$
end
foreach $l \in \mathcal{N}$ **do**
  $\mathbf{h}_l^0 \leftarrow [\mathbf{x}_l, 0 \ldots, 0]$
end
for $t = 1$ *to* T **do**
  foreach $p \in \mathcal{R}$ **do**
    foreach $l \in p$ **do**
      $\mathbf{h}_p^t \leftarrow RNN_t(\mathbf{h}_p^t, \mathbf{h}_l^t)$
      $\tilde{\mathbf{m}}_{p,l}^{t+1} \leftarrow \mathbf{h}_p^t$
    end
    $\mathbf{h}_p^{t+1} \leftarrow \mathbf{h}_p^t$
  end
  foreach $l \in \mathcal{N}$ **do**
    $\mathbf{m}_l^{t+1} \leftarrow \sum_{p:l \in p} \tilde{\mathbf{m}}_{p,l}^{t+1}$
    $\mathbf{h}_l^{t+1} \leftarrow U_t\left(\mathbf{h}_l^t, \mathbf{m}_l^{t+1}\right)$
  end
end
$\hat{\mathbf{y}}_p \leftarrow F_p(\mathbf{h}_p)$

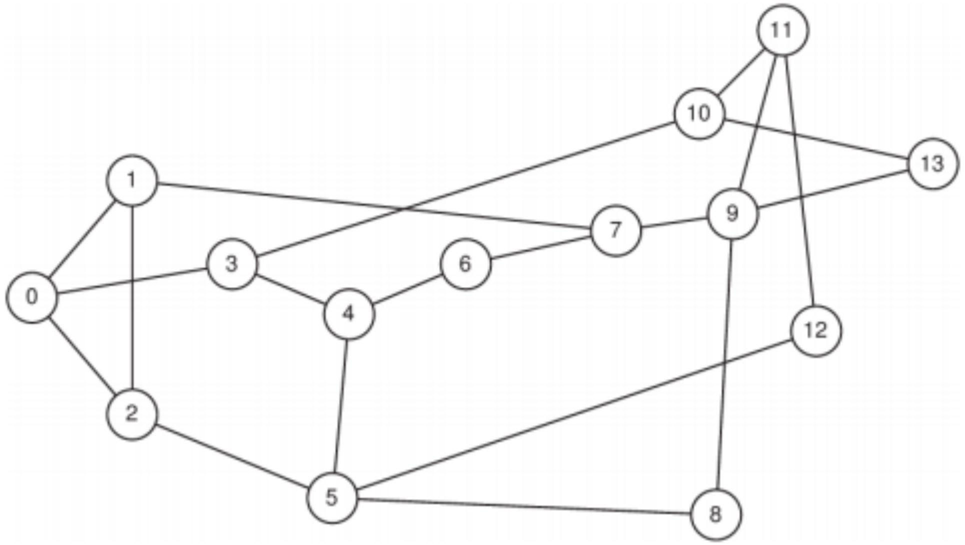
Graph Neural Networks are **not a black-box** and require a **custom architecture** for each problem we are modeling. This needs to be done by a **ML & Networking expert.**

$$\mathbf{h}_{l_i} = f(\mathbf{h}_{p_1}, \ldots, \mathbf{h}_{p_j}), \quad l_i \in p_k, k = 1, \ldots, j$$
$$\mathbf{h}_{p_k} = g(\mathbf{h}_{l_{k(0)}}, \ldots, \mathbf{h}_{l_{k(|p_k|)}})$$

How accurate is RouteNet?

RouteNet: Dataset

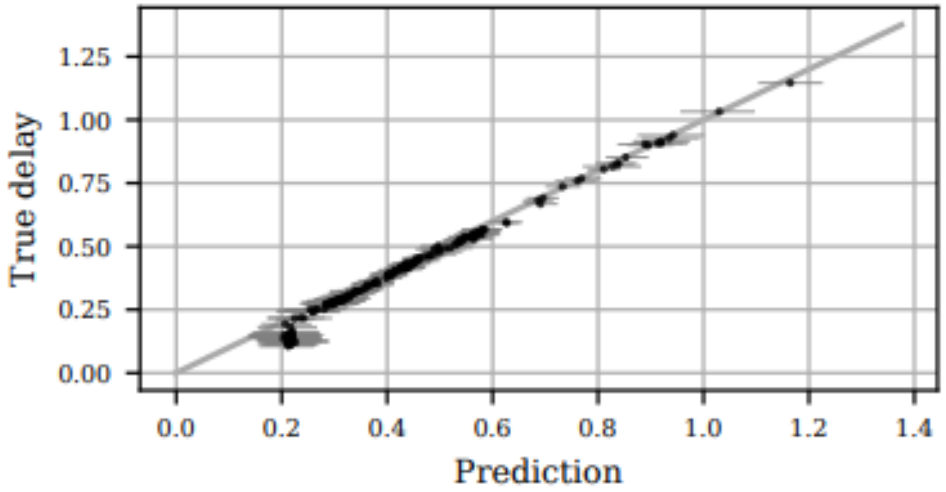
- Dataset obtained with simulation
  - Omnet++
  - Event per-packet simulator that considers queuing
- Trained with the NSFnet 14-node topology
- 260k samples of random (uniform)
  - Traffic Matrices
  - Routing Configurations
  - Resulting per-packet average delay, jitter and losses



NSFnet Topology

RouteNet: Accuracy

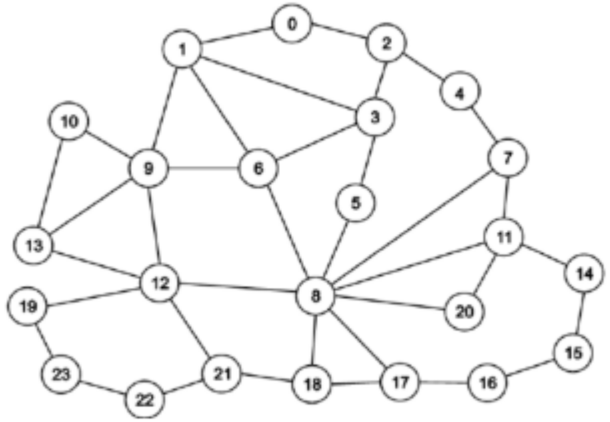
	NSF	
	Delay	**Jitter**
R^2	0.99	0.98
ρ	0.998	0.993



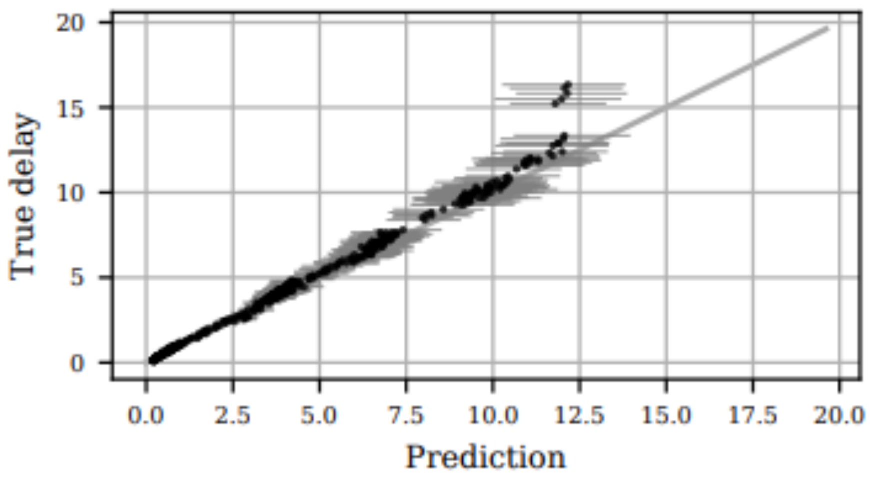
- RouteNet achieves good accuracy
- The Readout is use as dropout to prevent overfitting
- Transfer learning is used to improve learning for jitter and drops.

RouteNet: Generalization

	NSF		Geant2	
	Delay	Jitter	Delay	Jitter
R^2	0.99	0.98	0.97	0.86
ρ	0.998	0.993	0.991	0.942



Geant Topology

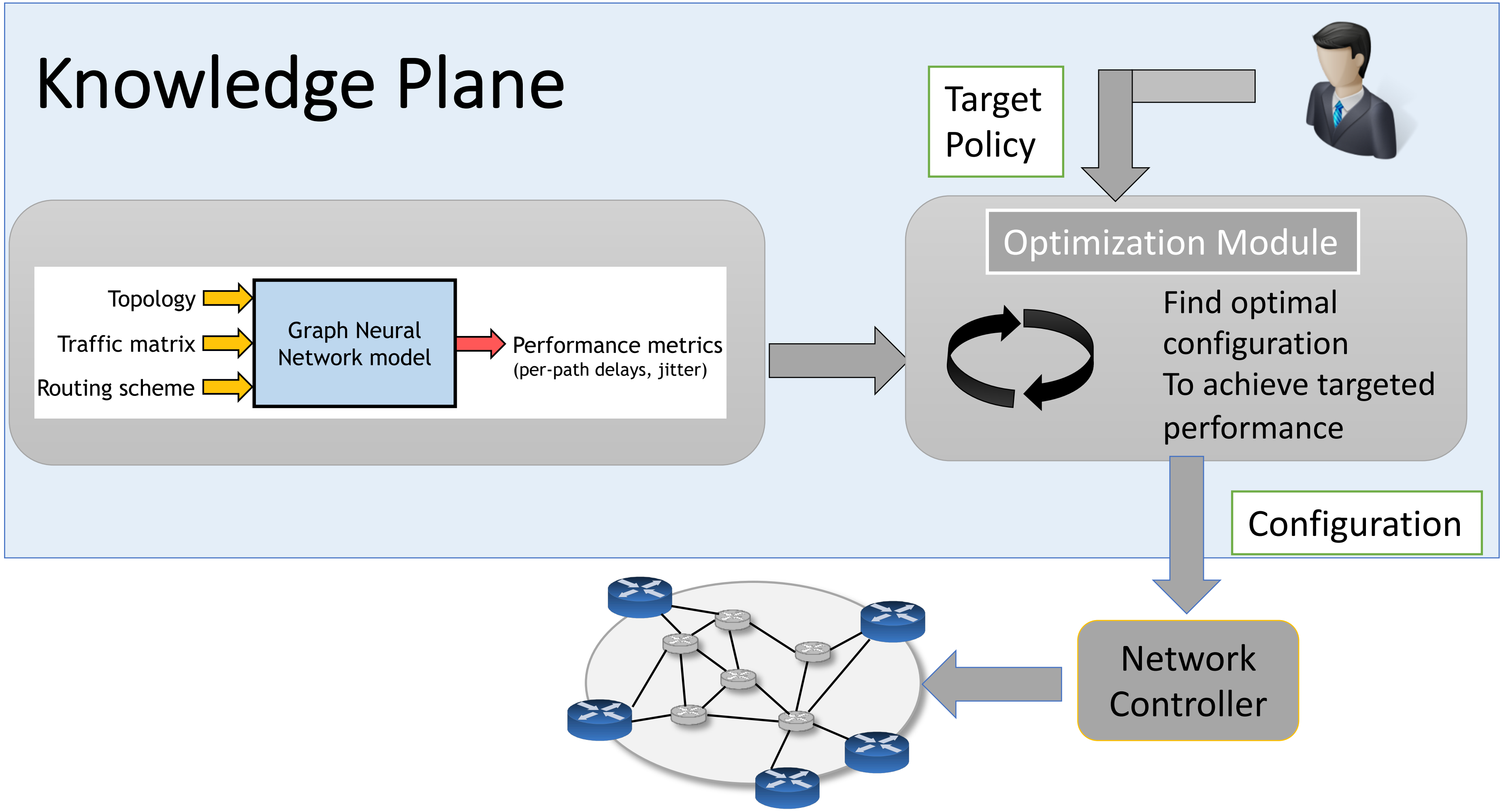


- What happens when we evaluate RouteNet with an **unseen** topology?
- We tested RouteNet with the 24-node Geant topology
- RouteNet produces accurate estimates for an unseen topologies.

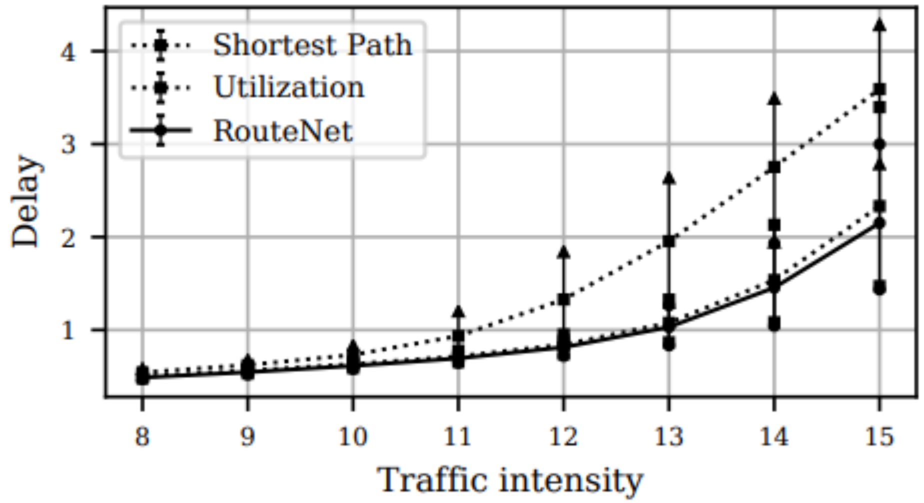
RouteNet can ***generalize*** *to unseen* topologies, routings and traffic matrices.

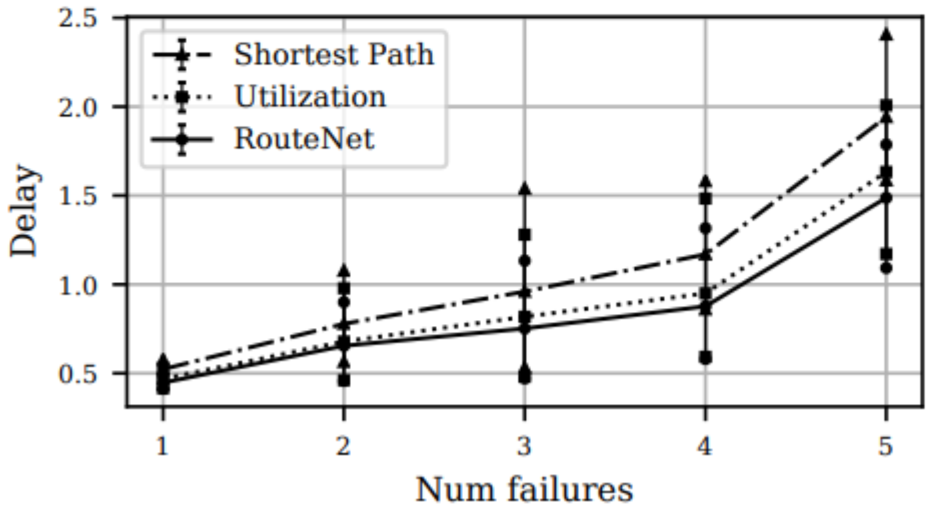
How can we use RouteNet for Netwok Optimization?

Network Optimization Architecture



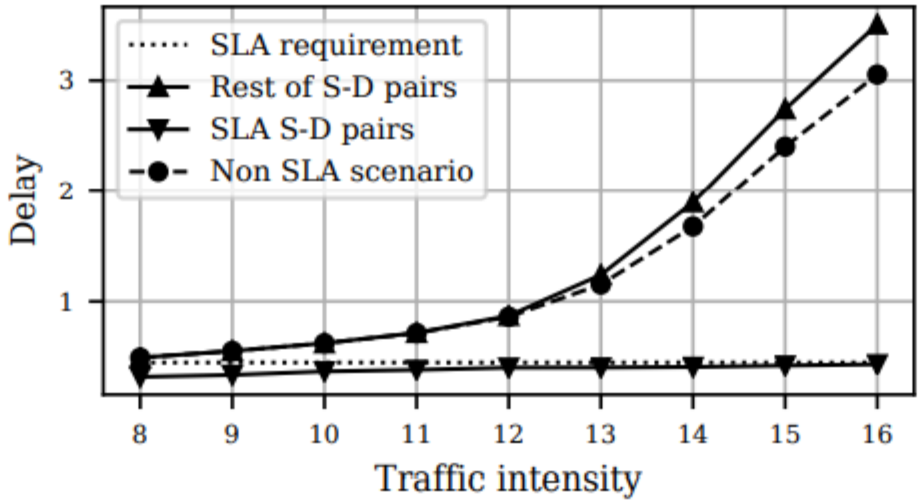
RouteNet: Delay/Jitter Routing Optimization





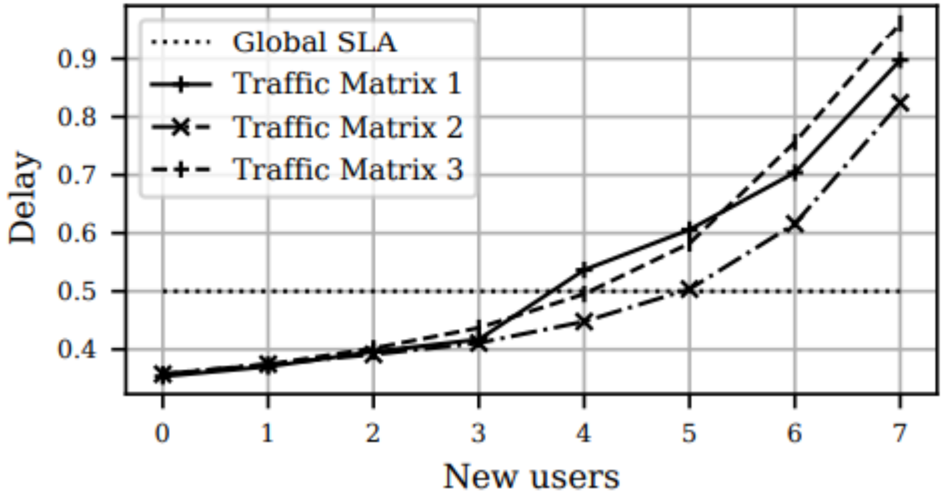
- **Find routes that optimize maximum delay/jitter/drops for each source-destination pair in the network.**
- In the presence of link-failures
- Only consider variations of shortest-path routing
- Compaired against common approaches

RouteNet: SLA optimization



- **Keep delay/jitter/drops below a certain SLA treshold for premium users in the presence of growing overall traffic.**
- Only consider variations of shortest-path routing
- Compaired against common approaches

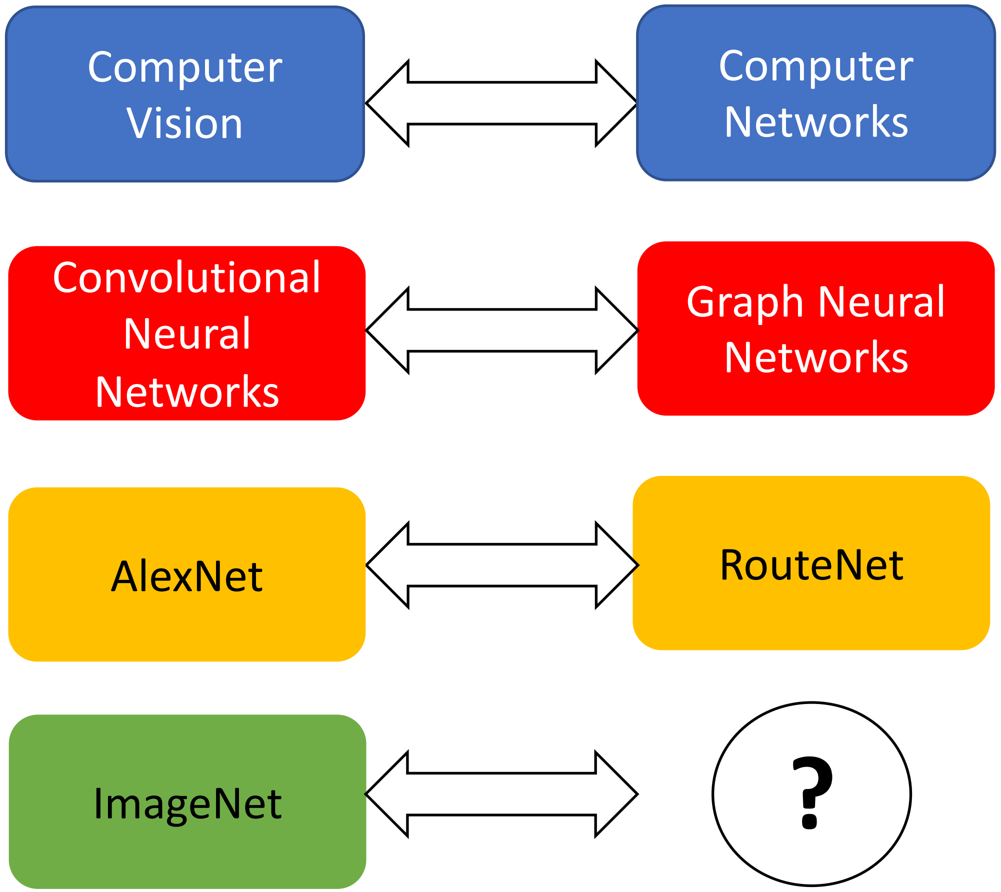
RouteNet: Network Planning



Traffic matrix	Optimal new link placement	Previous delay	Delay with new link	Delay reduction
TM_1	(1, 9)	0.732	0.478	35.7 %
TM_2	(2, 13)	0.996	0.464	53.4 %
TM_3	(1, 9)	1.179	0.516	56.2 %
TM_4	(2, 11)	0.966	0.518	46.37 %
TM_5	(1, 11)	0.908	0.502	44.7 %
TM_6	(0, 13)	0.811	0.484	40.3 %
TM_7	(1, 12)	0.842	0.485	42.4 %
TM_8	(1, 11)	0.770	0.431	44.0 %
TM_9	(1, 9)	1.009	0.492	51.2 %
TM_{10}	(2, 11)	1.070	0.491	54.1 %

- **When to upgrade the network?**
  - Given a certain organic growth of users, when the delay/jitter/losses will be above a certain threshold?
- **What link to upgrade?**
  - What is the optimal link to upgrade to improve performance?

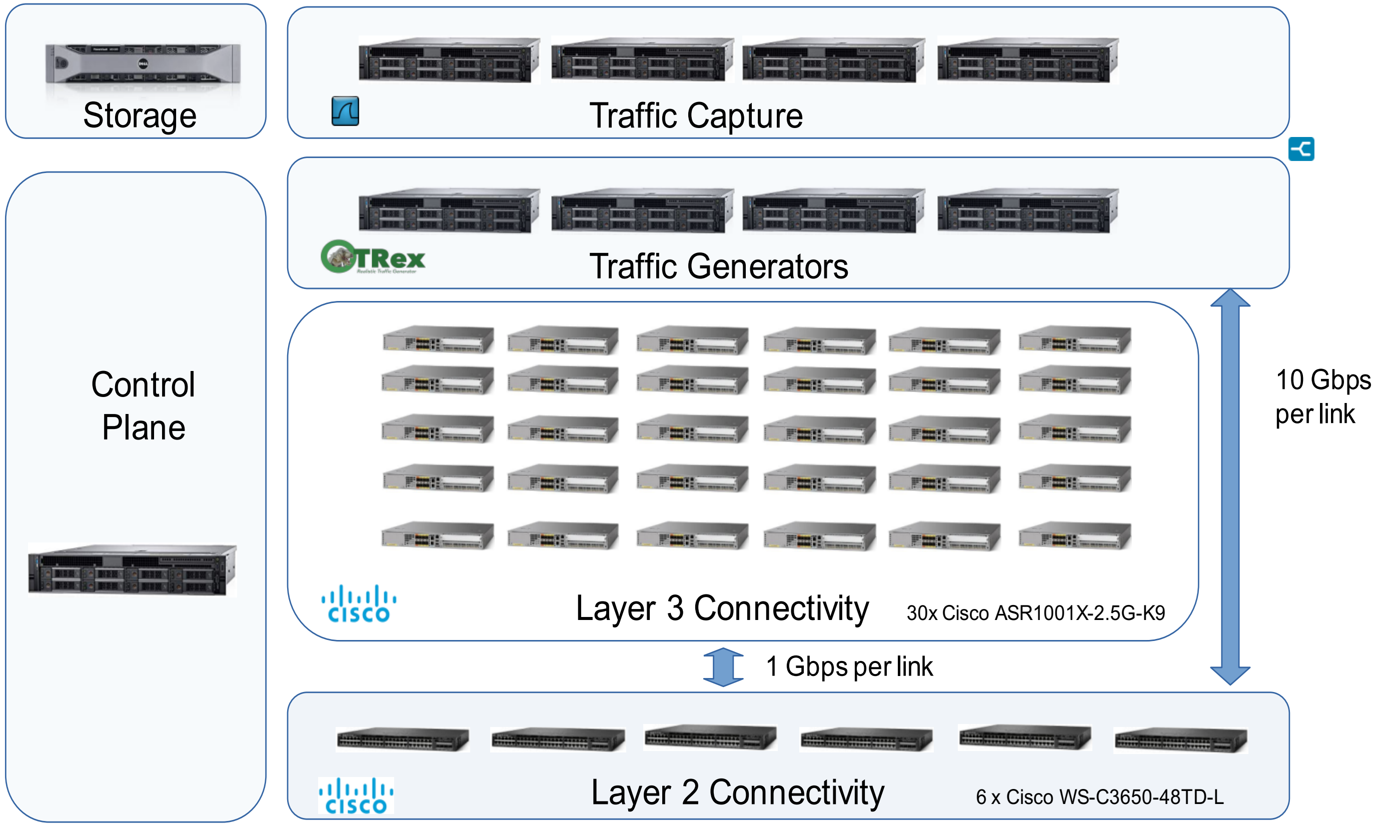
Graph Neural Networking



- CNNs are to Computer Vision what GNNs to Computer Networks
- RouteNet represents AlexNet
- But we don´t have ImageNet?

We need a data-set for Computer Networks

DataNet: A labeled data-set for Network ML



- At the Barcelona Neural Networking Center we want to build **DataNet**
- This is the **enabler** of research in Graph Neural Networks
- An experimental test-bed to produce datasets for ML
- Real-world networks with evolving complexity

Barcelona Neural Networking Center

Barcelona Neural Networking Center

Barcelona Neural Networking Center

UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

- Our goal is to become the **enablers** in the application of ML to networking

1. **DataNet** → An experimental testbed to produce data-sets for ML applied to Networking
2. **RouteNet** → An open-source implementation of GNN for Computer Networks