Advancing Packet-Level Traffic Predictions with Transformers

Siddhant Ray

D-ITET
ETH Zürich

September 1, 2022

Networked Systems
ETH Zürich — seit 2015

Problems with machine learning methods in networks today

Problems with machine learning methods in networks today

- No generalization: Only work on specific tasks trained on

Problems with machine learning methods in networks today

- No generalization: Only work on specific tasks trained on
- Limited scope: Models fail outside original training environment

Problems with machine learning methods in networks today

- No generalization: Only work on specific tasks trained on
- Limited scope: Models fail outside original training environment
- Resource intensive: Always re-doing training from scratch

Why use Transformers?

Why use Transformers?

- Efficient learning with *attention* mechanism
- Generalizing using *large datasets* available

Why use Transformers?

- Efficient learning with *attention* mechanism
- Generalizing using *large datasets* available
- State-of-art for *sequence* learning problems
- Network *packet data* is a sequence

Transformer's unprecendented generalization in NLP & CV

Transformer's unprecendented generalization in NLP & CV

BERT: Generalizing to many tasks in NLP

- Sentiment analysis
- Question answering
- Paraphrase detection

Transformer's unprecendented generalization in NLP & CV

BERT: Generalizing to many tasks in NLP

- Sentiment analysis
- Question answering
- Paraphrase detection

Vision Transformer: Generalizing to many tasks in CV

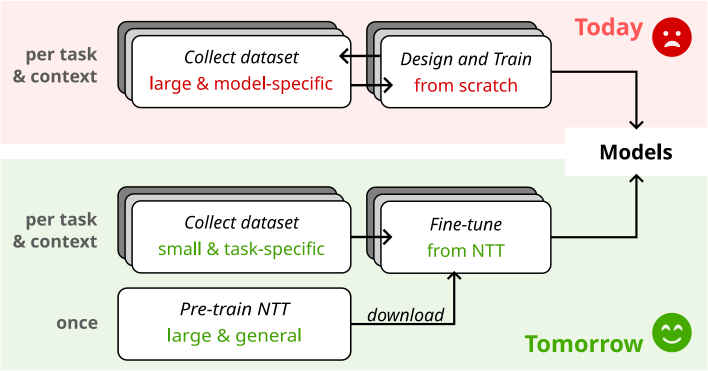
- Image classification
- Object detection
- Image segmentation

Our Transformer prototype

We present our Network Traffic Transformer (NTT):

Our Transformer prototype

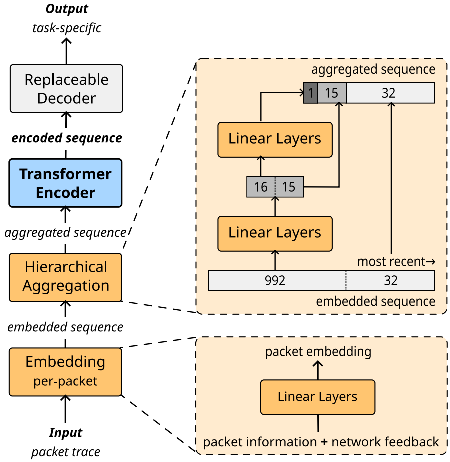
We present our Network Traffic Transformer (NTT):



Pre-train today, fine-tune and re-use tomorrow

NTT needs networking domain specific features

NTT needs networking domain specific features



The Network Traffic Transformer (NTT) with an embedding layer, an aggregation layer, a transformer encoder and a task-specific replaceable decoder.

NTT needs networking domain specific features

Feature selection for initial NTT's input data:

NTT needs networking domain specific features

Feature selection for initial NTT's input data:

- Relative timestamp: To learn sequence order
- End-to-end delay: To learn network state information
- Packet size: To learn packet state information

Training objectives for the NTT architecture

NTT's learning objectives:

Training objectives for the NTT architecture

NTT's learning objectives:

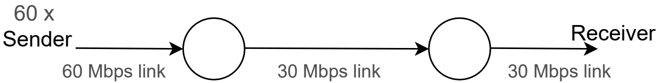
- Learn network dynamics: Reconstruct masked delay values

Training objectives for the NTT architecture

NTT's learning objectives:

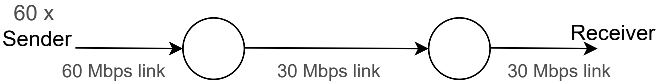
- Learn network dynamics: Reconstruct masked delay values
- Scale to large sequences: Aggregate inputs ($> 1000s$ of values)

Ensuring varied dynamics in our pre-training datasets



Initial topology for data generation

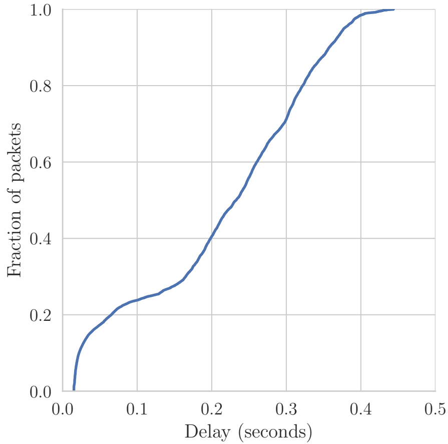
Ensuring varied dynamics in our pre-training datasets



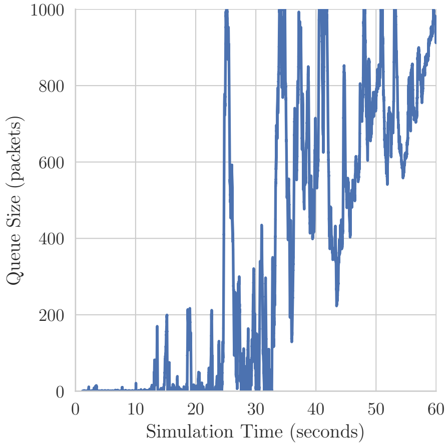
Initial topology for data generation

- Varied start times across sender application flows
- Enough variance in pre-training data dynamics

Ensuring varied dynamics in our pre-training datasets



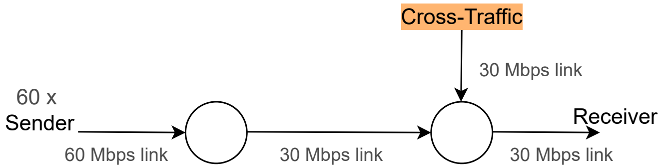
Delay CDF, single simulation run



Bottleneck queue profile on the single-path topology

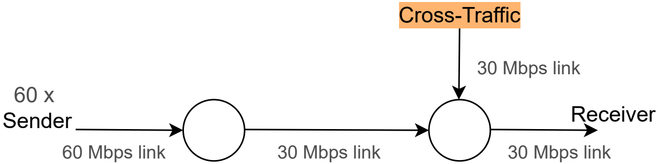
Distribution plots on pre-training data

Our NTT allows for generalization on network dynamics



Fine-tuning data generation, single path topology

Our NTT allows for generalization on network dynamics



Fine-tuning data generation, single path topology

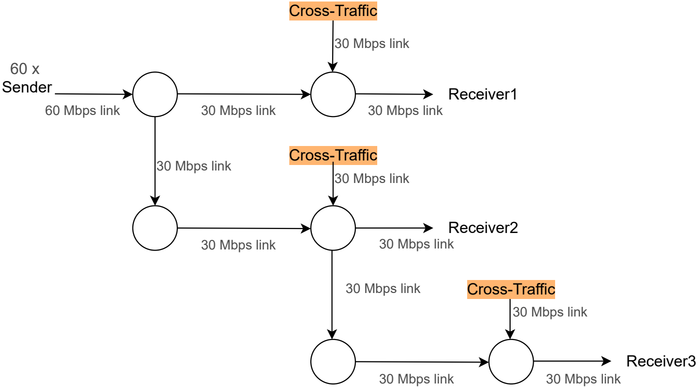
- Two bottleneck dynamics to learn
- Packet-level fine-tuning task : Predict last delay
- Flow-level fine-tuning task : Predict Message Completion Time (MCT)

Our NTT allows for generalization on network dynamics

all values $\times 10^{-3}$	Pre-training	Fine-tuning	
	Delay	Delay	log (MCT)
NTT			
Pre-trained	**0.072**	**0.097**	**65**
From scratch	-	0.313	117
Baselines			
ARMA	1.800	1.180	1412
Last observed	0.142	0.121	2189
EWMA	0.259	0.211	1147
NTT (Ablated)			
No aggregation	0.258	0.430	61
Fixed aggregation	0.055	0.134	115
Without packet size	0.001	8.688	94
Without delay	15.797	10.898	802

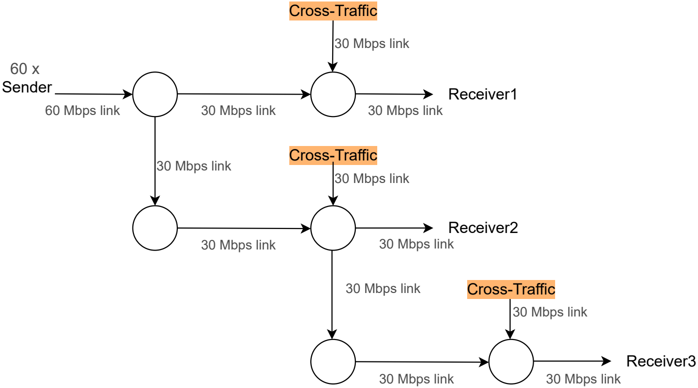
Mean Squared Error (MSE) for all NTT models and tasks for the single path topology (lower is better)

NTT works on multi-path topologies



Fine-tuning data generation on multi-path topology

NTT works on multi-path topologies



Fine-tuning data generation on multi-path topology

- Path delays vary as per number of links
- Receiver ID as IP address proxy

NTT works on multi-path topologies

Model	MSE: Delay Prediction *all values*$\times 10^{-3}$	# of Epochs trained
NTT		
Pre-trained + Fine-tune (full)	**0.004**	**5**
Pre-trained + Fine-tune (10%)	**0.035**	**12**
From scratch + Fine-tune (full)	5.2	10
From scratch + Fine-tune (10%)	8.2	15
Baselines		
ARMA	4.2	-
Last observed	11.2	-
EWMA	4.0	-
NTT (Ablated)		
Pre-trained + Fine-tune (full) : No Receiver ID	2.8	8
From scratch + Fine-tune (full) : No Receiver ID	2.7	15

Fine-tuning NTT on the multi-path topology (lower is better)

How do we improve the NTT further?

How do we improve the NTT further?

- NTT Scaling
  - Learn additional network features.
  - Learn on larger topologies.

How do we improve the NTT further?

- NTT Scaling
  - Learn additional network features.
  - Learn on larger topologies.
- Federated Learning
  - Share models, not data.
  - Keep data private.

How do we improve the NTT further?

- NTT Scaling
  - Learn additional network features.
  - Learn on larger topologies.
- Federated Learning
  - Share models, not data.
  - Keep data private.
- Continual learning
  - Re-train with time, prevent forgetting.
  - Learn evolved dynamics.

Recap: Our NTT architecture demonstrates that

Recap: Our NTT architecture demonstrates that

- Learning network dynamics is possible
  - NTT learns network dynamics from packet sequences
  - Pre-trained NTT can be re-used easily

Recap: Our NTT architecture demonstrates that

- Learning network dynamics is possible
  - NTT learns network dynamics from packet sequences
  - Pre-trained NTT can be re-used easily
- Generalizing power of the NTT
  - Can generalize to new environments: Packet level
  - Can generalize to new tasks: Flow level