Analysis of Flow Prolongation Using Graph Neural Network in FIFO Multiplexing System

6 December 2022

Master Degree Project Student: Weiran Wang

EPFL Supervisor: Hossein Tabatabaee

KTH Examiner: Prof. Viktoria Fodor

EPFL Examiner: Prof. Jean-Yves Le Boudec

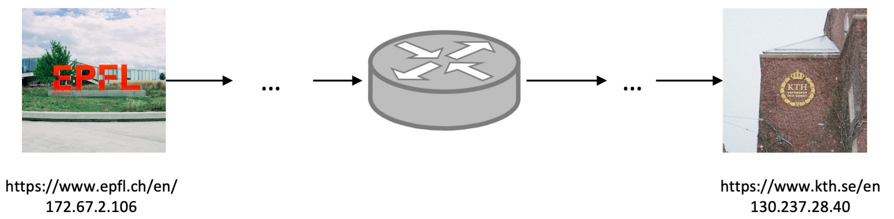
Project Motivations

- In a network setting, computing the tightest delay bound is hard, even in a FIFO network
- Network Calculus provides a mathematical framework and several approaches to calculate the delay bound
  - Unfortunately, these delay bounds are usaully not tight
- Flow prolongation has been found to be potential to tighten the delay bound
  - Finding the best flow prolongation combinations is hard due to the scalability
- Graph Neural Network (GNN) is used to find the best flow prolongation combinations to tighten the delay bound
  - Both flow prolongation and GNN are pioneering in the field of Network Calculus
- The robustness and accuracy of the GNN model needs to be benchmarked

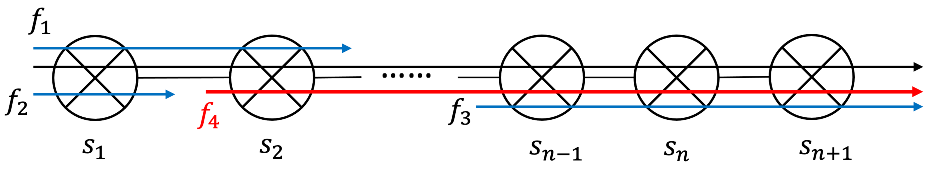
Introduction

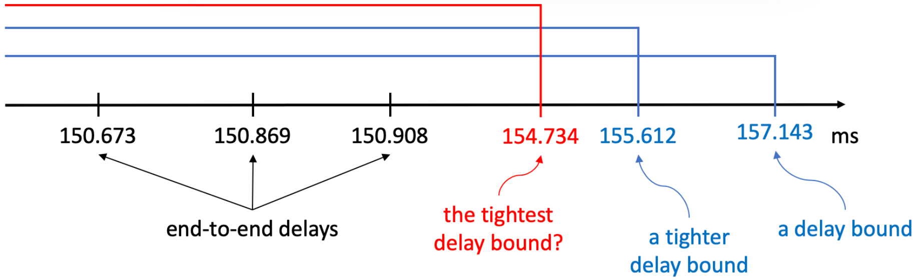
Delay Bound

- Delay Bound is an upper bound of the worst-case end-to-end delay
- The flow whose end-to-end delay needs to be analyzed is defined as the flow of interest
- Finding the tightest delay bound is defined as NP-hard




```
PING www.kth.se (130.237.28.40): 56 data bytes
64 bytes from 130.237.28.40: icmp_seq=0 ttl=247 time=15.213 ms
64 bytes from 130.237.28.40: icmp_seq=1 ttl=247 time=20.152 ms
64 bytes from 130.237.28.40: icmp_seq=2 ttl=247 time=17.488 ms
64 bytes from 130.237.28.40: icmp_seq=3 ttl=247 time=196.816 ms
64 bytes from 130.237.28.40: icmp_seq=4 ttl=247 time=3.278 ms
64 bytes from 130.237.28.40: icmp_seq=5 ttl=247 time=17.992 ms
64 bytes from 130.237.28.40: icmp_seq=6 ttl=247 time=21.790 ms
64 bytes from 130.237.28.40: icmp_seq=7 ttl=247 time=54.969 ms
^C
--- www.kth.se ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.278/43.462/196.816/59.581 ms
```



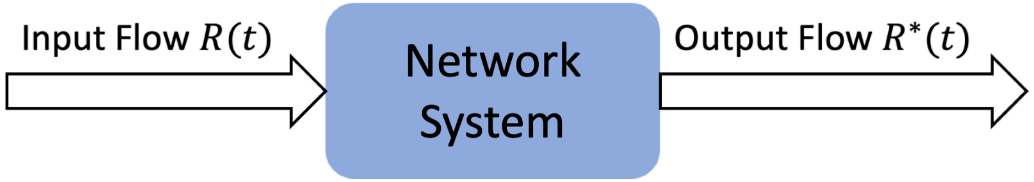


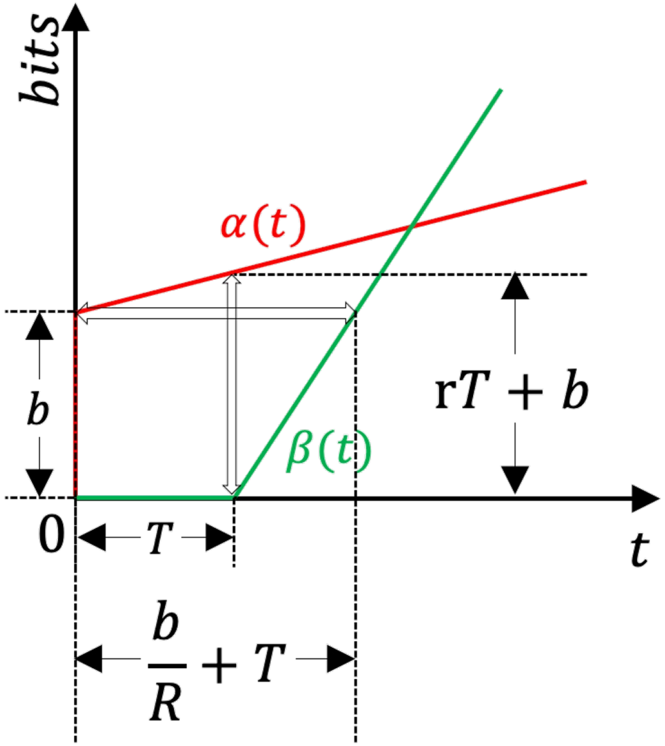
Bouillard, Anne, Laurent Jouhet, and Eric Thierry. "Tight performance bounds in the worst-case analysis of feed-forward networks." *2010 Proceedings IEEE INFOCOM*. IEEE, 2010.

Introduction

Network Calculus

- View a network system as a queuing framework
- Provide a series of mathematical functions for finding an upper bound of an end-to-end delay
- Arrival Curve $\alpha(t)$ (determined by flow rate and burst) generated by the flows limits the bits entering the system
- Service Curve $\beta(t)$ (determined by server rate and latency) offered by the network system guarantees the Quality of Service to the flows
- Network Calculus uses these two curves to compute the delay bound, namely the largest horizontal deviation

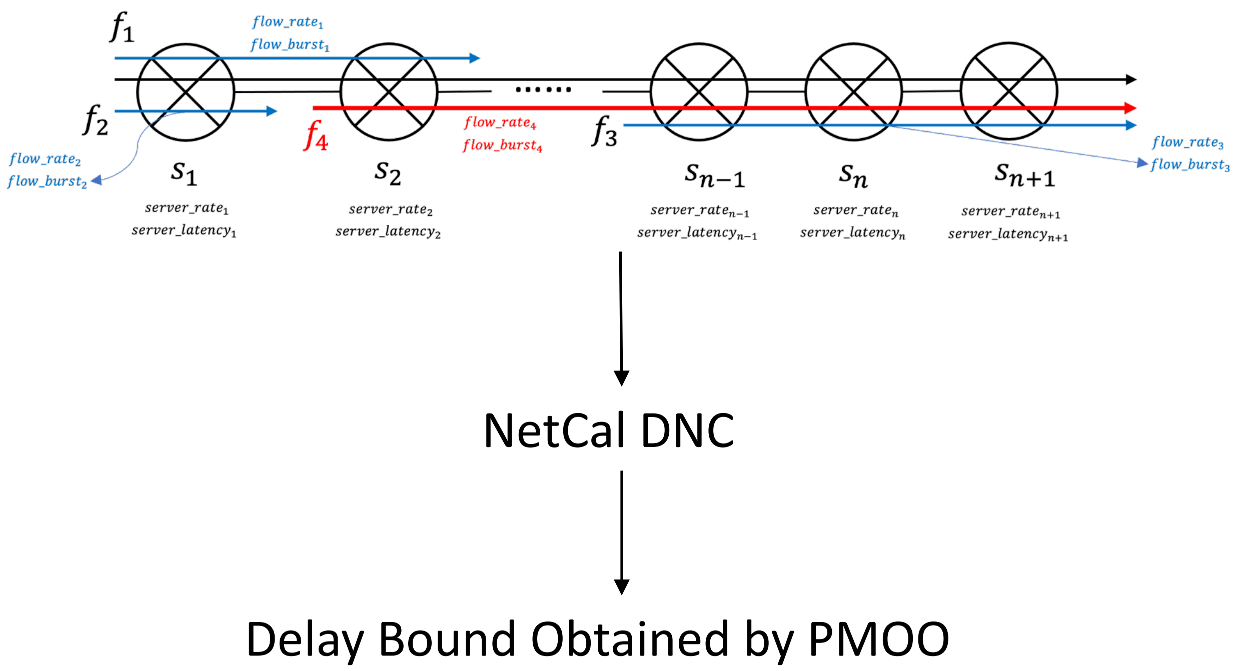




Introduction

Delay Bound Calculation Method

- Various delay bound calculation methods are investigated by scientists in various years, e.g., TMA, SFA, PMOO, LUDB, DEBORAH
- Leads to different tightnesses and different execution times
- Pay Multiplexing Only Once (PMOO) is used due to its good trade-off between tightness and exection time
- NetCal DNC, an open source software to calculate the delay bound, is chosen in this project



Bondorf, Steffen, and Jens B. Schmitt. "The DiscoDNC v2: A comprehensive tool for deterministic network calculus." *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*. 2014.

Introduction

Flow Prolongation Definition

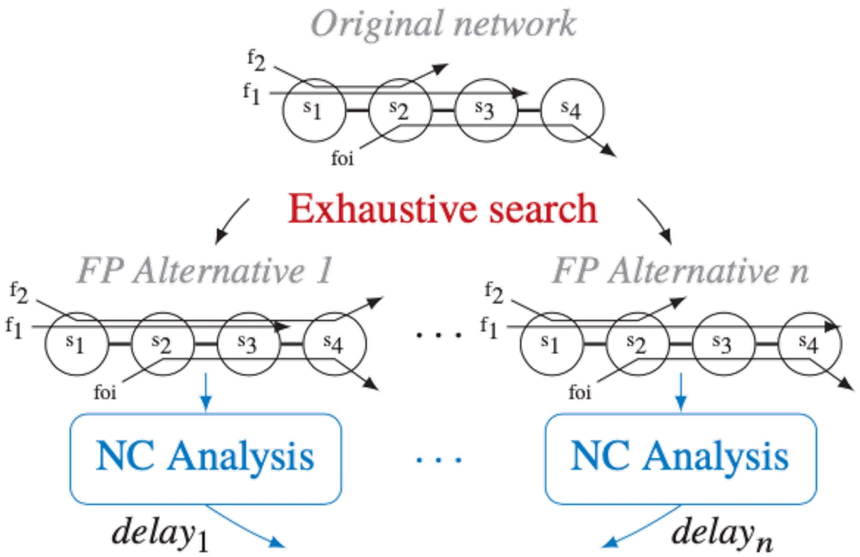
- Potentially tighten the delay bound obtained by PMOO
- Extend the path of cross flows to a new sink server
- The path of flow of interest will not be prolonged
- The most accurate and rigorous way is by exhaustive search

$$O(n^m)$$

n: # servers

m: # cross flows

Not scalable if implemented in exhaustive search



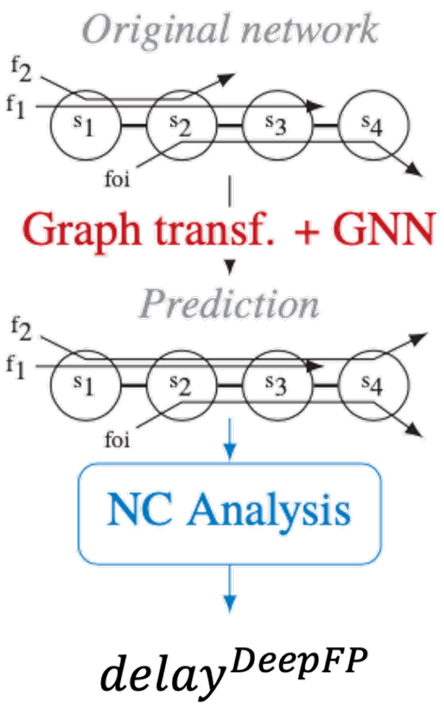
$$delay^{FP} = \min(delay_1, delay_2, \dots, delay_n)$$

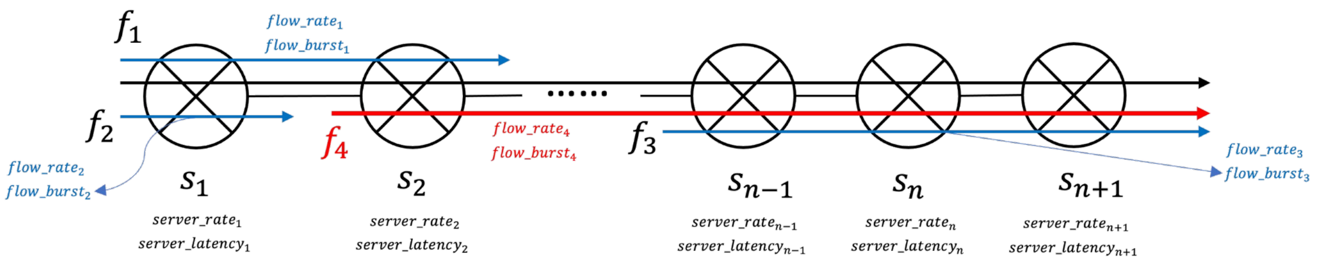
Geyer, Fabien, Alexander Scheffler, and Steffen Bondorf. "Tightening Network Calculus Delay Bounds by Predicting Flow Prolongations in the FIFO Analysis." *Proceedings of the 27th IEEE Real-Time and Embedded Technology and Applications Symposium*. 2021.

Introduction

Flow Prolongation with Machine Learning

- GNN is trained based on network features (rate-latency servers, token-bucket flows, flow of interest)
- The best prolongation combinations in the dataset is found by exhaustive searches beforehand





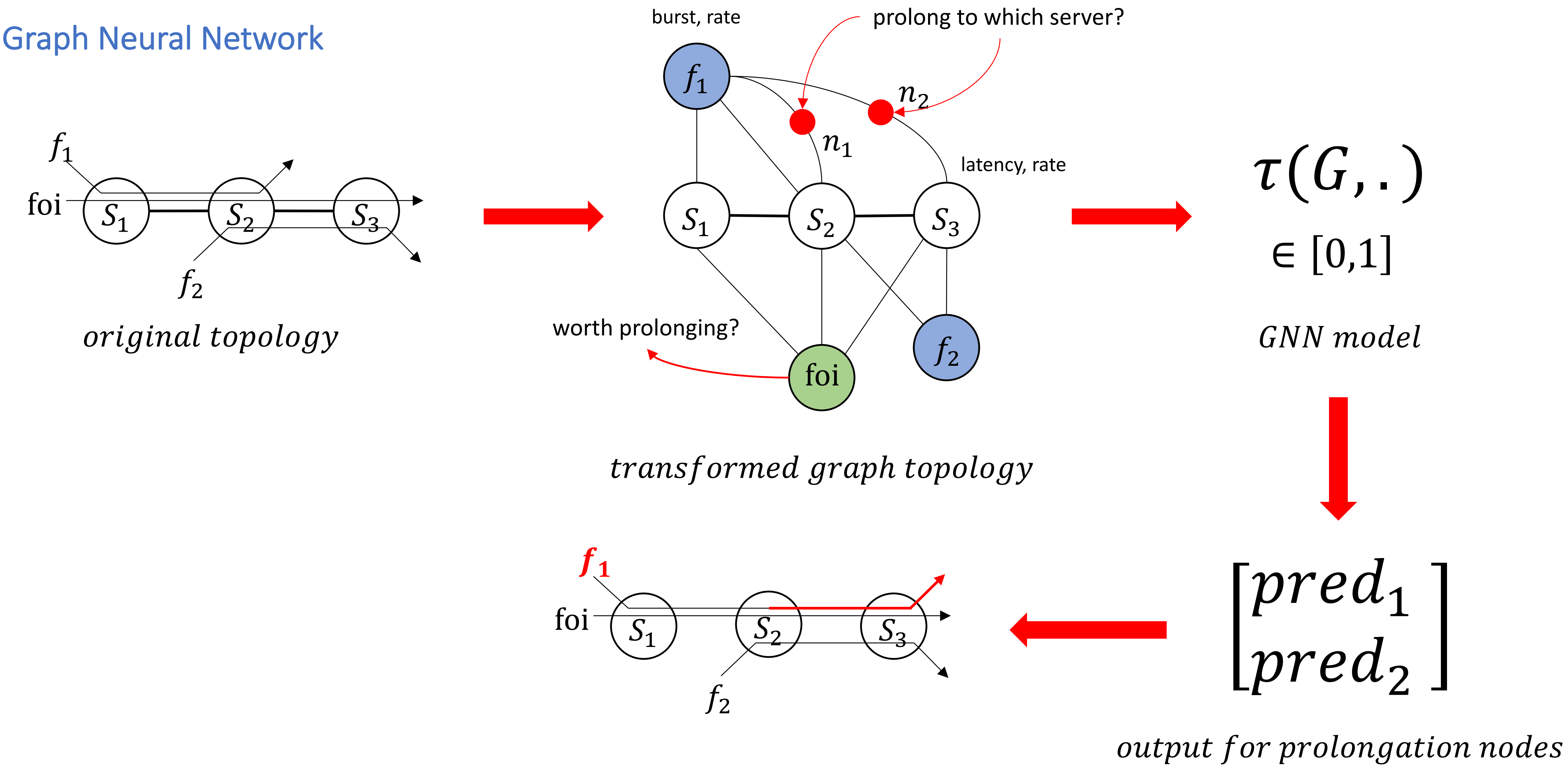
Parameter	Min	Max	Mean
# of servers	4	10	7.8
# of flows	5	35	24.5
# of cross-flows	1	21	4.1
# of prolong. comb. (PMOO-FP$_{foi}$)	2	4024	16.8
# of prolong. comb. (DEBORAH-FP$_{foi}$)	2	131072	247.1
Flow path length	3	9	4.1
Number of nodes in graph	11	128	43.3

datasets parameters used to train the GNN model

Geyer, Fabien, Alexander Scheffler, and Steffen Bondorf. "Tightening Network Calculus Delay Bounds by Predicting Flow Prolongations in the FIFO Analysis." *Proceedings of the 27th IEEE Real-Time and Embedded Technology and Applications Symposium*. 2021.

Introduction

Graph Neural Network



Introduction

GNN Outputs

*reproduced deepfp on **PMOO** accruacy: 65%*

(69.6% in the paper)

1. **pred1**: Decide if it is worthwhile to apply the prolongation algorithm on this flow of interest scenario (threshold = 0.5)
2. **pred2**: Decide where to prolong the flows if necessary (criteria: the highest value)

foi	start server	sink server	PRED1 before attack
6	3	1	0.9892914891242980
flow id	**start server**	**sink server**	**PRED2 before attack**
1	2	2	1.0
1	2	1	8.28888158110885E-09
2	7	3	0.2397686094045640
2	7	2	0.7816663384437560
2	7	1	0.004808166529983280
4	2	2	1.0
4	2	1	7.38276773049051E-09
7	2	2	1.0
7	2	1	8.52992521060969E-09
12	4	2	0.9865729808807370
12	4	1	0.00841361004859209

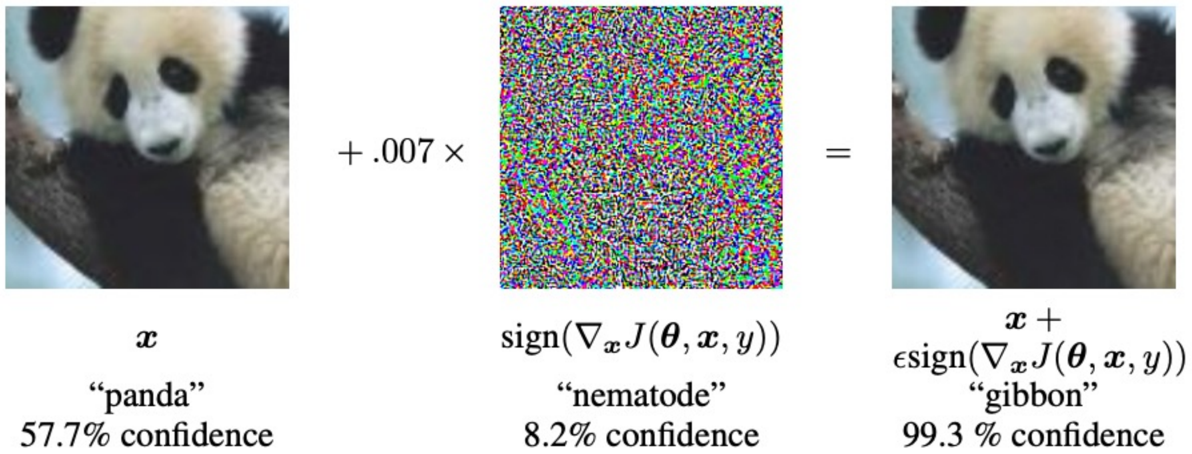
an example of the foi 6 in the 0^{th} topology in the open source dataset

Geyer, Fabien, Alexander Scheffler, and Steffen Bondorf. "Tightening Network Calculus Delay Bounds by Predicting Flow Prolongations in the FIFO Analysis." *Proceedings of the 27th IEEE Real-Time and Embedded Technology and Applications Symposium*. 2021.

Goal

Adversarial Attack

- The robustness of the machine learning model has been attracting lots of attentions in recent years
- By modifying the inputs a little bit, the outputs of machine learning will be quite different
- Fast Gradient Sign Method (FGSM) is used in this project



Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Main Tasks

Available Tools:

- NetCal/DNC written in Java
- Pre-trained GNN code based on DEBORAH to predict the best prolonged topologies

Tasks Done:

- Modified the GNN code and trained a new model based on PMOO
- Integrated NetCal/DNC into GNN so that the delay bound can be calculated for a given network topology
- Based on the GNN prediction results, found the potential attack targets
- Realized the FGSM adversarial attack under the project background
- Created a larger dataset for the adversarial attack purpose
- Analyzed the adversarial attack results, i.e., tested whether GNN is fooled to predict the wrong flow prolongation, and thus lossen the delay bound

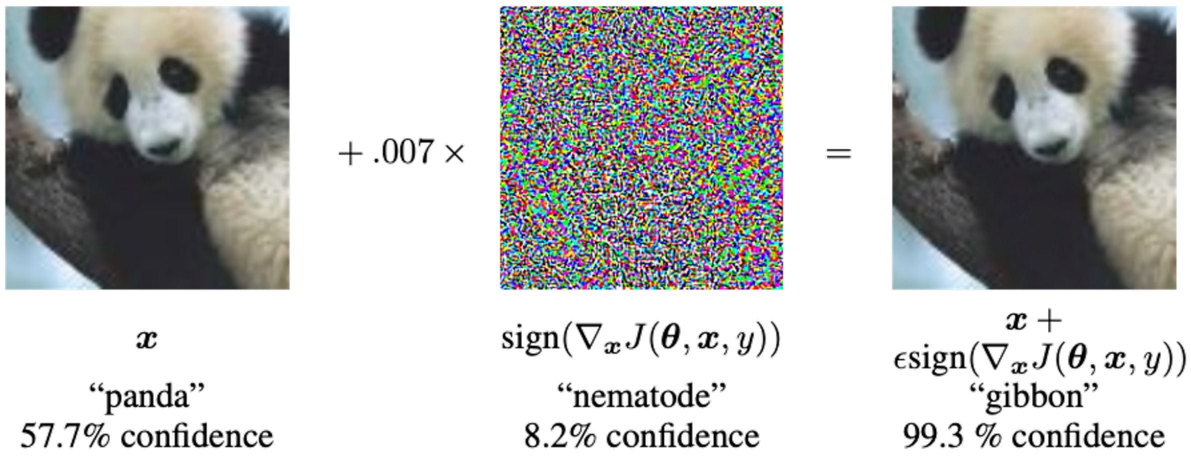
Bondorf, Steffen, and Jens B. Schmitt. "The DiscoDNC v2: A comprehensive tool for deterministic network calculus." *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*. 2014.

Methods

Fast Gradient Sign Method

$$\hat{x} = x + \varepsilon \cdot \operatorname{sign}(\nabla_x J(\theta, x, y))$$

- $\boldsymbol{x}$: Input data (server rates and latency, flow rates and bursts) in our case
- $\boldsymbol{\theta}$: GNN model weights
- $\boldsymbol{y}$: the correct flow prolongations given by the dataset (found by exhaust search)
- $\boldsymbol{J(\theta, x, y)}$: loss function of applying the GNN with parameters θ and datapoint(x, y)
- $\boldsymbol{sign(a)}$: $\begin{cases} \mathbf{1}, \boldsymbol{a} > \mathbf{0} \\ \mathbf{0}, \boldsymbol{a} = \mathbf{0} \\ \mathbf{-1}, \boldsymbol{a} < \mathbf{0} \end{cases}$
- $\boldsymbol{\varepsilon}$: perturbed factor
- $\hat{\boldsymbol{x}}$: perturbed input data

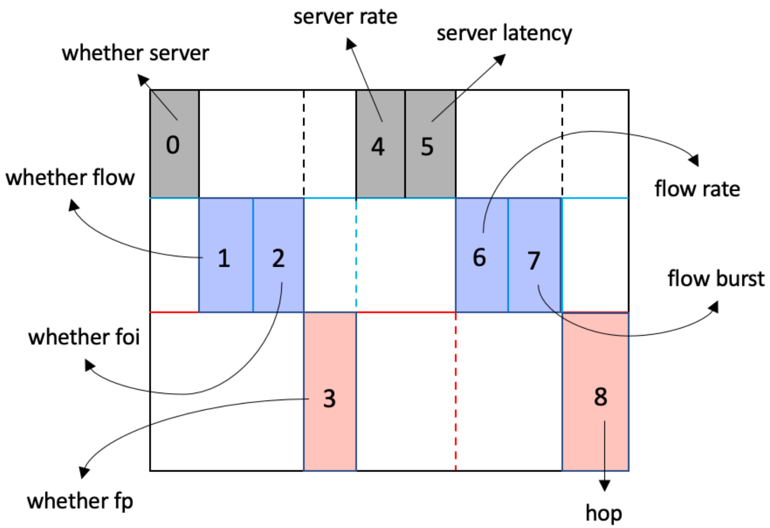


Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Methods

FGSM Implementation in Network Features

$$\hat{x} = x + \varepsilon \cdot \operatorname{sign}(\nabla_x J(\theta, x, y))$$

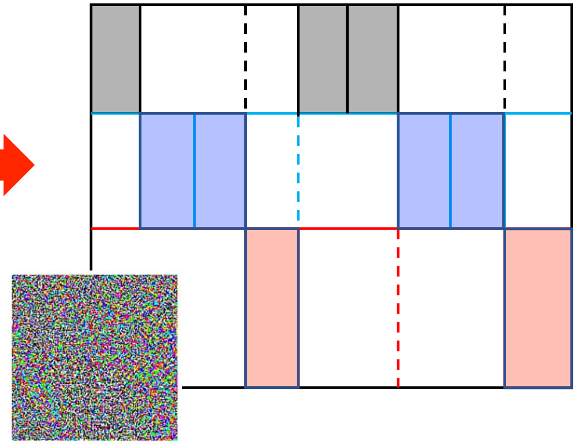


$original\ netowrk\ features$

GNN

$DeepFP$

$J(\theta, x, y)$

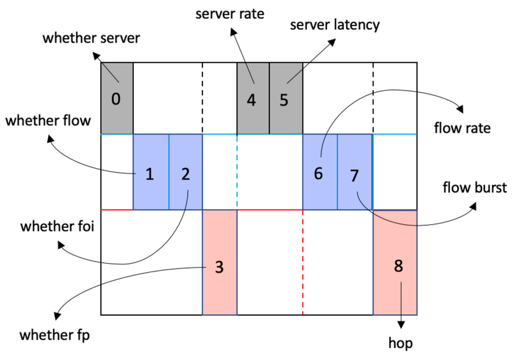


$attacked\ netowrk\ features$

$\varepsilon \in [0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.011, 0.012, 0.013, 0.014, 0.015, 0.016, 0.017, 0.018, 0.019, 0.02]$

Methods

Fast Gradient Sign Method

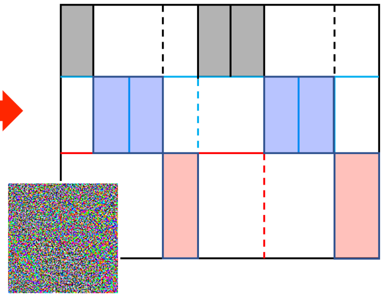


original netowrk features

GNN

DeepFP

$J(\theta, x, y)$



attacked netowrk features

$$\hat{x} = x + \varepsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

$$\boldsymbol{\hat{x} = torch.clamp(\hat{x})}$$

- replace the server rate/latency with the minimum server rate/latency in this topology if the value after the attack is smaller than 0
- replace the server rate/latency with the maximum server rate/latency in this topology if the value after the attack is larger than 1
- same with the flow rate/burst

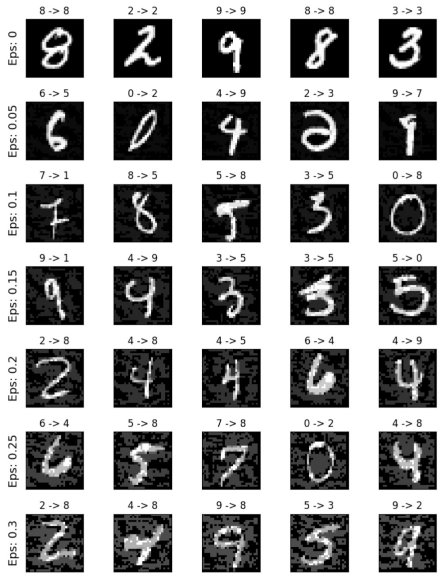
Methods

Larger Dateset Creation Motivation

- Analyzed on the open-source dataset, but the results were far from satisfying
- Guessed that it might be the small size of network leading to the non-obvious attack results
- Imitated the Computer Vision

Parameter	Min	Max	Mean
# of servers	4	10	7.8
# of flows	5	35	24.5
# of cross-flows	1	21	4.1
# of prolong. comb. (PMOO-FP$_{\text{foi}}$)	2	4024	16.8
# of prolong. comb. (DEBORAH-FP$_{\text{foi}}$)	2	131072	247.1
Flow path length	3	9	4.1
Number of nodes in graph	11	128	43.3

datasets parameters used to train the GNN model



Changeable Network Features

- min: $4 \cdot 2 + 5 \cdot 2 = 18$
- max: $10 \cdot 2 + 35 \cdot 2 = 90$
- $mean$: $7.8 \cdot 2 + 24.5 \cdot 2 = 64.6$

MNIST pixel:

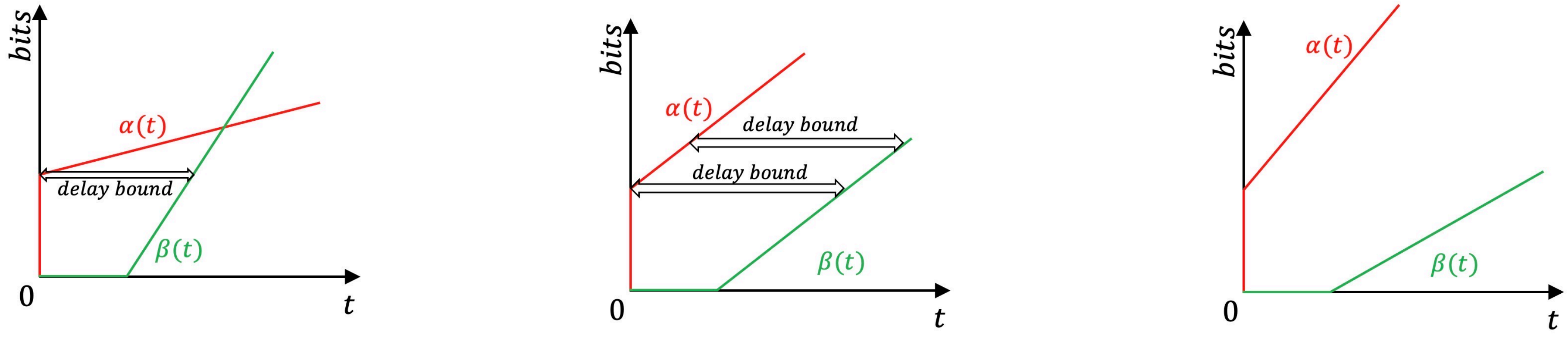
Each image is a crude $28 \cdot 28 = 784$ pixels digit

LeCun, Yann. "The MNIST database of handwritten digits." *http://yann. lecun. com/exdb/mnist/* (1998).

Methods

Larger Dateset Creation Criteria

- The aggregated flow rate on one server should not exceed this server rate
- Each flow path is unique without redundancy
- The source server of the foi is one of the three servers, and the sink server is the last server in the network topology
- Exhaustive search is used to find the tighter delay bounds
- Once three tighter delay bounds are found, the next network topology will be generated



$R > r$ $\quad$ $R = r$ $\quad$ $R < r$

Methods

Larger Dataset Creation Results

- Created a larger dateaset than the open-source one
- The number of network features are still far from figure pixels used in the Computer Vision

Parameter	Min	Max	Mean
# of servers	4	10	7.8
# of flows	5	35	24.5
# of cross-flows	1	21	4.1
# of prolong. comb. (PMOO-FP$_{\text{foi}}$)	2	4024	16.8
# of prolong. comb. (DEBORAH-FP$_{\text{foi}}$)	2	131072	247.1
Flow path length	3	9	4.1
Number of nodes in graph	11	128	43.3

datasets parameters used to train the GNN model

Parameter	Min	Max	Mean
# of servers	20	30	24.9
# of flows	46	232	115.6
Flow path length	1	30	9.3

newly created dataset with larger number of servers and flows used for the adversarial attack purpose

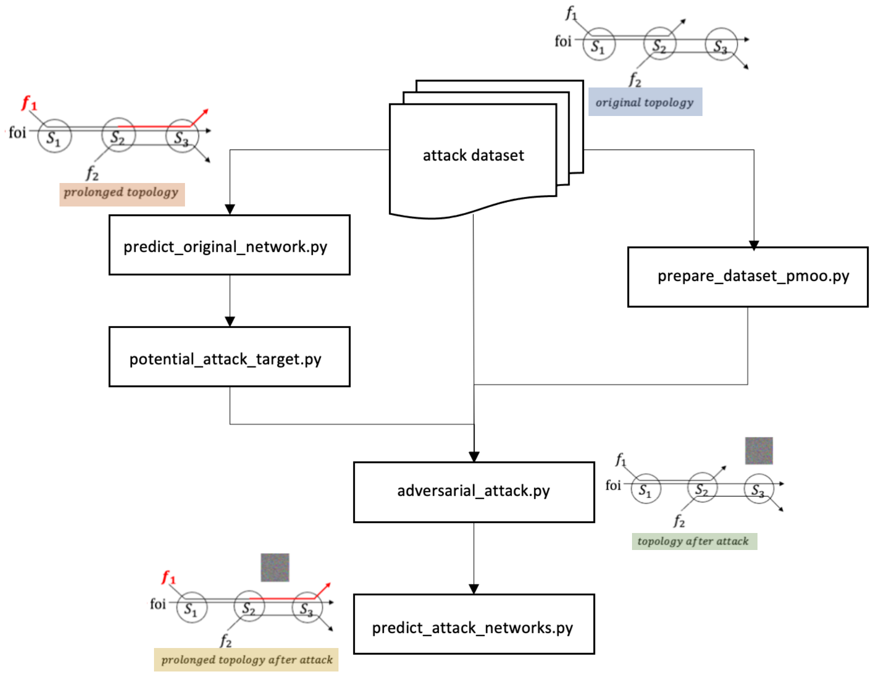
topo id	# server	# flow	server rate	server latency	flow rate	flow burst
0 - 999	20 - 30	n*(n+1)/5	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
1000 - 1499	30 - 40	n*(n+1)/5	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
1500 - 1999	40 - 50	n*(n+1)/5	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
2000 - 2499	20 - 30	n*(n+1)/4	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
2500 - 2999	20 - 30	n*(n+1)/3	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
3000 - 3499	20 - 30	n*(n+1)/6	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
3500 - 3999	20 - 30	n*(n+1)/7	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
4000 - 4499	20 - 30	n*(n+1)/8	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
4500 - 4999	20 - 30	n*(n+1)/9	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1
5000 - 5099	25	25*26/5=130	0.01	0.05	0.0005	0.05
5100 - 5199	25	25*26/5=130	0.02	0.05	0.0005	0.05
5200 - 5299	25	25*26/5=130	0.03	0.05	0.0005	0.05
5300 - 5399	25	25*26/5=130	0.04	0.05	0.0005	0.05
5400 - 5499	25	25*26/5=130	0.05	0.05	0.0005	0.05
5500 - 5599	25	25*26/5=130	0.06	0.05	0.0005	0.05
5600 - 5699	25	25*26/5=130	0.07	0.05	0.0005	0.05
5700 - 5799	25	25*26/5=130	0.08	0.05	0.0005	0.05
5800 - 5899	25	25*26/5=130	0.09	0.05	0.0005	0.05
5900 - 5999	25	25*26/5=130	0.1	0.05	0.0005	0.05
6000 - 6099	25	25*26/5=130	0.03	0.02	0.0005	0.05
6100 - 6199	25	25*26/5=130	0.03	0.04	0.0005	0.05
6200 - 6299	25	25*26/5=130	0.03	0.06	0.0005	0.05
6300 - 6399	25	25*26/5=130	0.03	0.08	0.0005	0.05
6400 - 6499	25	25*26/5=130	0.03	0.1	0.0005	0.05
6500 - 6599	25	25*26/5=130	0.03	0.12	0.0005	0.05
6600 - 6699	25	25*26/5=130	0.03	0.14	0.0005	0.05
6700 - 6799	25	25*26/5=130	0.03	0.16	0.0005	0.05
6800 - 6899	25	25*26/5=130	0.03	0.18	0.0005	0.05
6900 - 6999	25	25*26/5=130	0.03	0.2	0.0005	0.05
7000 - 7099	25	25*26/5=130	0.03	0.05	0.0001	0.05
7100 - 7199	25	25*26/5=130	0.03	0.05	0.0002	0.05
7200 - 7299	25	25*26/5=130	0.03	0.05	0.0003	0.05
7300 - 7399	25	25*26/5=130	0.03	0.05	0.0004	0.05
7400 - 7499	25	25*26/5=130	0.03	0.05	0.0005	0.05
7500 - 7599	25	25*26/5=130	0.03	0.05	0.0006	0.05
7600 - 7699	25	25*26/5=130	0.03	0.05	0.0007	0.05
7700 - 7799	25	25*26/5=130	0.03	0.05	0.0008	0.05
7800 - 7899	25	25*26/5=130	0.03	0.05	0.0009	0.05
7900 - 7999	25	25*26/5=130	0.03	0.05	0.001	0.05
8000 - 8099	25	25*26/5=130	0.03	0.05	0.0005	0.01
8100 - 8199	25	25*26/5=130	0.03	0.05	0.0005	0.02
8200 - 8299	25	25*26/5=130	0.03	0.05	0.0005	0.03
8300 - 8399	25	25*26/5=130	0.03	0.05	0.0005	0.04
8400 - 8499	25	25*26/5=130	0.03	0.05	0.0005	0.05
8500 - 8599	25	25*26/5=130	0.03	0.05	0.0005	0.06
8600 - 8699	25	25*26/5=130	0.03	0.05	0.0005	0.07
8700 - 8799	25	25*26/5=130	0.03	0.05	0.0005	0.08
8800 - 8899	25	25*26/5=130	0.03	0.05	0.0005	0.09
8900 - 8999	25	25*26/5=130	0.03	0.05	0.0005	0.1
9000 - 9999	20 - 30	n*(n+1)/5	0.05 - 1	0.01 - 1	0.0001 - 0.0005	0.01 - 1

different # servers
different # flows
different server rates
different server latencies
different flow rates
different flow bursts
attacks only on the flows

new dataset is created for the adversarial attack purpose

Methods

Recap the Attack Process and Naming Scheme



category	acronym	FP?	FGSM?
category 1	obfp	×	×
category 2	oafp	✓	×
category 3	abfp	×	✓
category 4	aafp	✓	✓

- obfp: original topology before flow prolongation and adversarial attack
- oafp: original topology after flow prolongation but before adversarial attack
- abfp: attacked topology before flow prolongation
- aafp: attacked topology after flow prolongation

$$|DelayBound_{abfp} - DelayBound_{obfp}| \Big/ DelayBound_{obfp} \rightarrow small$$

$$|DelayBound_{aafp} - DelayBound_{abfp}| \Big/ DelayBound_{abfp} \rightarrow large$$

Numerical Analysis

Two Representative Examples

$$\left|DelayBound_{abfp} - DelayBound_{obfp}\right| \Big/ DelayBound_{obfp} \rightarrow \color{red}{small}$$

$$\left|DelayBound_{aafp} - DelayBound_{abfp}\right| \Big/ DelayBound_{abfp} \rightarrow \color{red}{large}$$

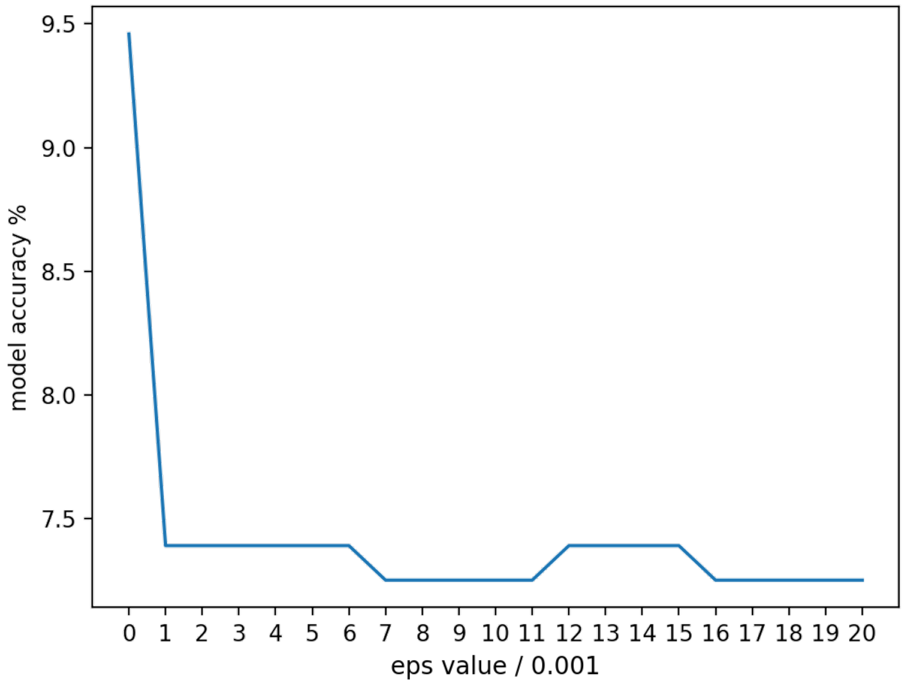
shown in the medium number

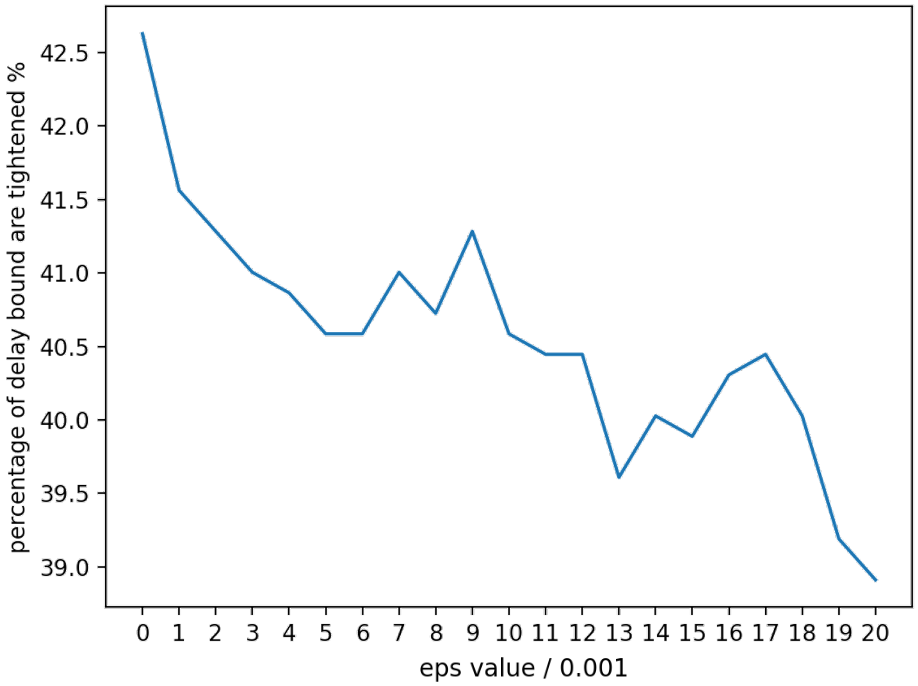
topo id	eps	delay bound obfp	delay bound oafp	delay bound abfp	delay bound aafp	server rate changes %	server latency chagnes %	flow rate changes %	flow burst changes %
6549	0.004	4659.031387	4651.304693	3840.871363	11106.02527	13.33332707	3.333336115	4.74975E-06	7.999999821
6369	0.002	7061.983539	7021.565914	8627.424152	18339.27296	6.666659315	2.499993518	2.666672319	3.999999166

- For topo id = 6549, eps=0.004
  - $\left|DelayBound_{abfp}-DelayBound_{obfp}\right| \Big/ DelayBound_{obfp} = 17.56\%$
  - $\left|DelayBound_{aafp}-DelayBound_{oafp}\right| \Big/ DelayBound_{oafp} = 189.15\%$
- For topo id = 6369, eps=0.002
  - $\left|DelayBound_{abfp}-DelayBound_{obfp}\right| \Big/ DelayBound_{obfp} = 22.17\%$
  - $\left|DelayBound_{aafp}-DelayBound_{oafp}\right| \Big/ DelayBound_{oafp} = 112.57\%$

Numerical Analysis

Model Accuracy and Tightened Networks Ratio





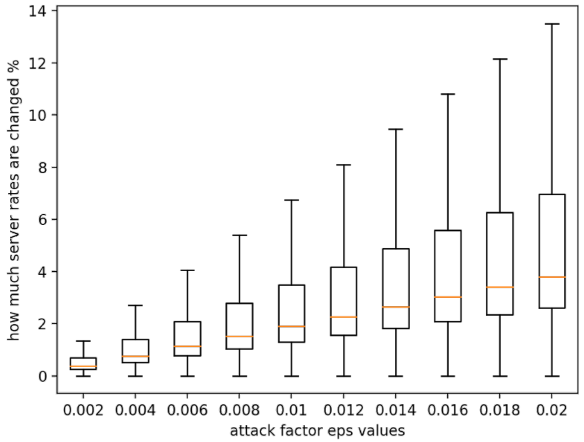
CONDITION: The prolonged topology after GNN is exactly the same shape with the target, i.e., the prolonged topolgy with the tightest delay bound stored in the dataset

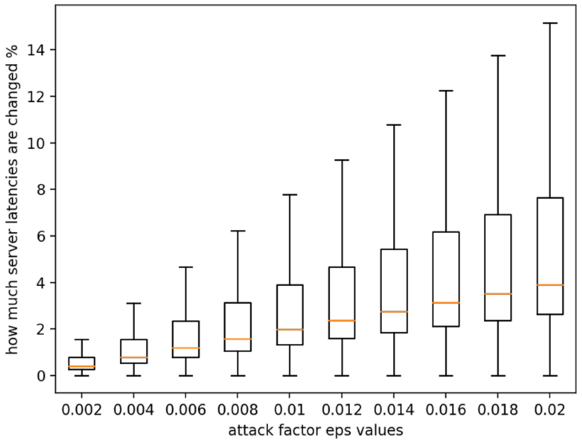
CONDITION: the delay bound after GNN prediction is tightened or is slightly larger than delay bound before the GNN prediction

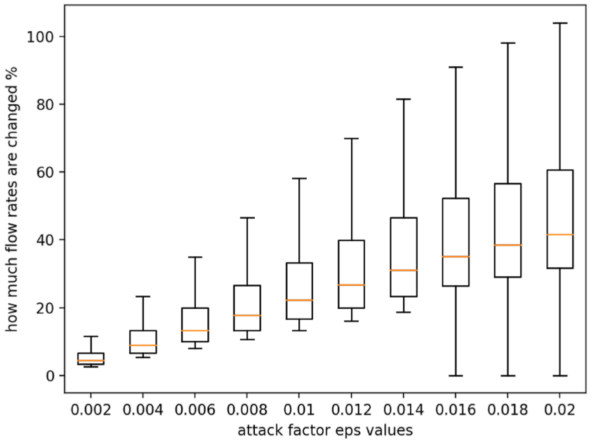
Geyer, Fabien, Alexander Scheffler, and Steffen Bondorf. "Network Calculus with Flow Prolongation--A Feedforward FIFO Analysis enabled by ML." *arXiv preprint arXiv:2202.03004* (2022).

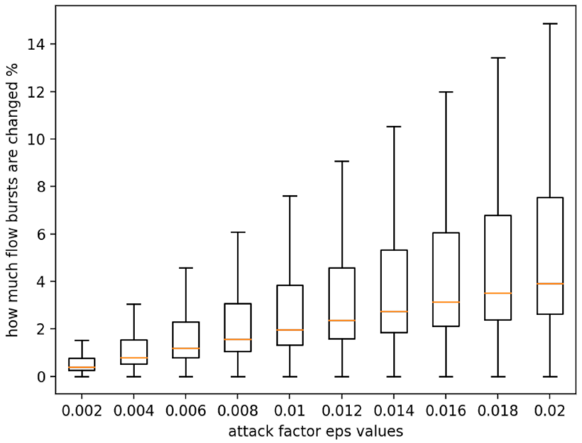
Numerical Analysis

Network Features Changes









CONDITION:

$$\frac{|network\ features\ after\ FGSM\ - network\ features\ before\ FGSM|}{network\ features\ before\ FGSM}$$

Numerical Analysis

Successful Attack Definition

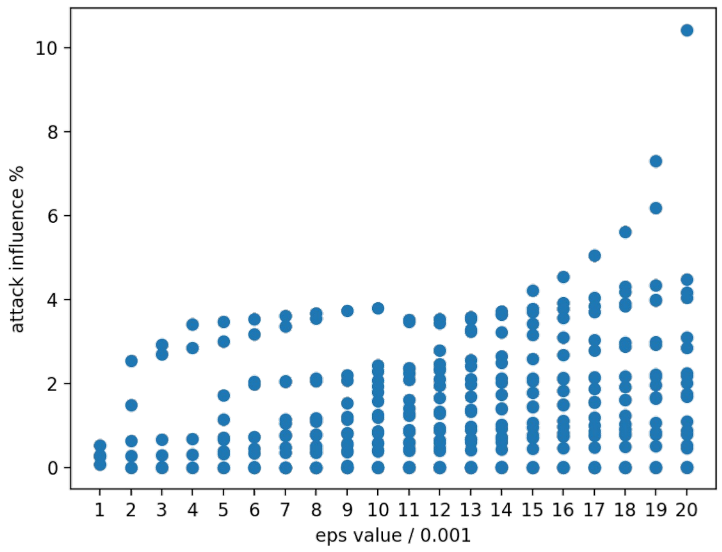
1. $DelayBound_{oafp} < DelayBound_{obfp}$

2. $DelayBound_{aafp} > DelayBound_{abfp}$

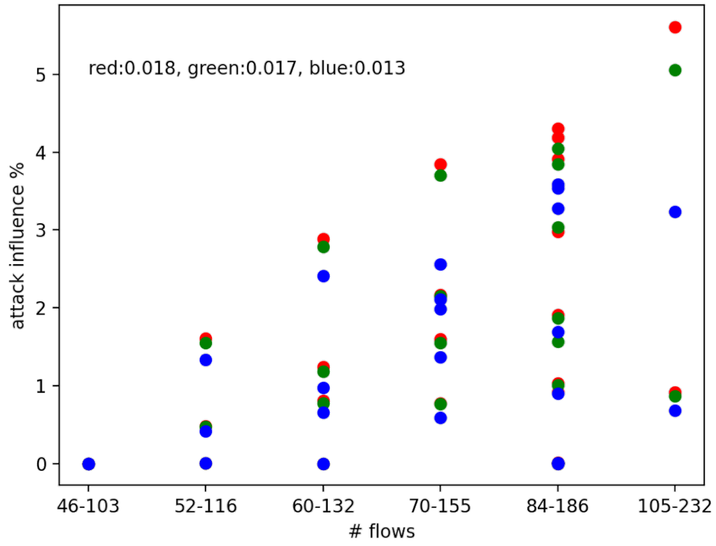
3. $^{|DelayBound_{abfp} - DelayBound_{obfp}|}/_{DelayBound_{obfp}} < 25\%$

ATTACK INFLUENCE:

$$^{|DelayBound_{aafp} - DelayBound_{abfp}|}/_{DelayBound_{abfp}}$$



except for some obvious examples, most of the attack influences locate between (0%, 10%)



a denser network tends to a larger attack influence

NETWORK SPARSITY:

$$\#server : n \in [20 \sim 30]$$

$$\#flow : \frac{n \cdot (n+1)}{m}, (m \in [4 \sim 9])$$

Conclusion

Tasks Done in this Project:

- Reproduced the GNN model based on PMOO and achieved an accuracy of 65% compared to 69.6% in the paper
- Integrated the NetCal DNC into the network topology so that once a new network is given, the delay bound can be calculated automatically
- Created a dataset with larger number of servers and flows inside for the adversarial attack pupose

Attack Results Summary:

- GNN or more generally speaking, machine learning models are first used for predicting flow prolongations and calculating the potential tightened delay bounds
- Current machine learning models are still under the stage for the smaller size of networks
- More than 160000 topologies have been analyzed
- The server rate, server latency and flow bursts are modified at max 14% after the attack, except for the flow rate, which is very sensitive to a little perturbation
- After defining the successful attack, except for some evident observations, most delay bounds after the attack and after the GNN flow prolongation are up to 10%
- The more sparse a network is, the larger attack influence value can be observed

Future Work

- If the industry can provide the dataset for the larger size of network, a new machine learning model based on this larger dataset can be trained, and a new benchmark can be done for the FGSM attack
- If the first bullet point succeeds, one more further step in the adversarial attack can be explored, e.g., a new model trained by Generative Adversarial Network (GAN) is proposed
- A defense procedure needs to be implementation to prevent the adversarial attack
- It is worth investigating which network features mainly influence the attack performance
- GNN can be used in other fields of the computer network, e.g., source allocation and scheduling problems

Azimi-Abarghouyi, Seyed Mohammad, and Viktoria Fodor. "Multi-Server Over-the-Air Federated Learning." *arXiv preprint arXiv:2211.16162* (2022).

Thank you for your listening