

On Traffic Prediction in Backbone Networks for Adaptive Proactive Protection

Attila Dobai-Pataky¹

Balázs Vass^{1,2}
Lehel Csató¹

June 9-11, 2025
RNDM, Trondheim

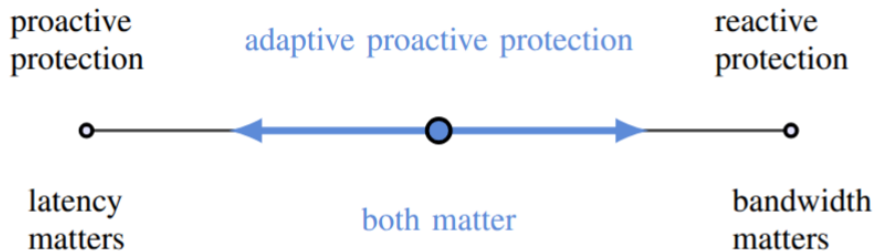
1: Babeş-Bolyai University, Cluj, Romania

2: Budapest University of Technology and Economics, Hungary

Quality of Service enhancement with Resilient routing and Machine learning

- QoS enhancement $\Rightarrow$ (in our case) increasing the availability (uptime) of a network
- with Resilient Routing
- and Machine learning

Reactive vs Proactive approaches



Resilient routing: A simple example

- Very important packet from *A* to *D*...

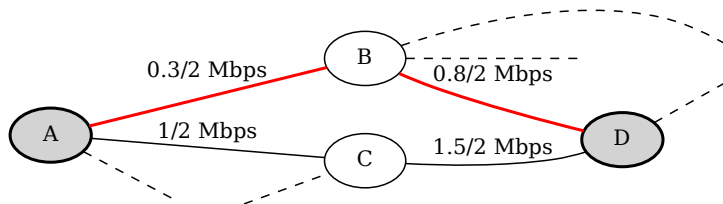


Figure 1: Possible routes in a network

Resilient routing: A simple example

- Very important packet from *A* to *D*...

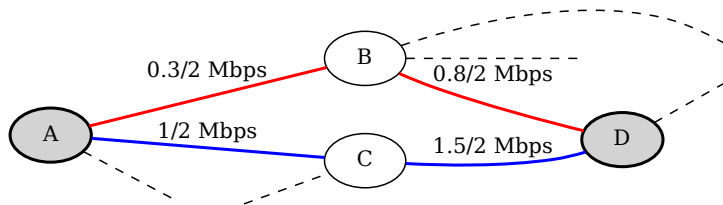


Figure 2: Possible routes in a network

- Send it on a backup route too!

The problem

Which backup route will have the required capacity?

- Idea: predict the free capacity **for each edge** – in a short time-distance.

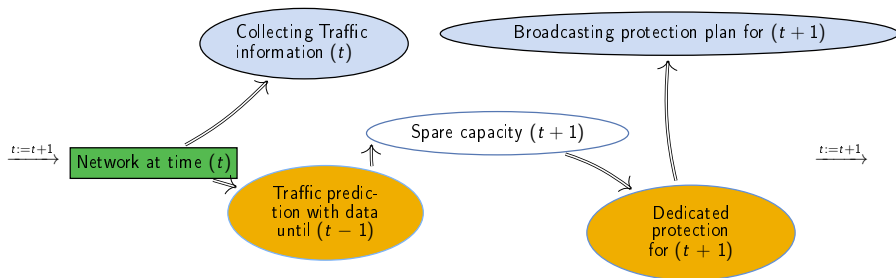


Figure 3: Quality of Service enhancement framework

Energy Sciences Network – ESNet

- ~ 141 links
- Bandwidth usage data, 30sec resolution
- A previous study: Mogyorósi et al.¹

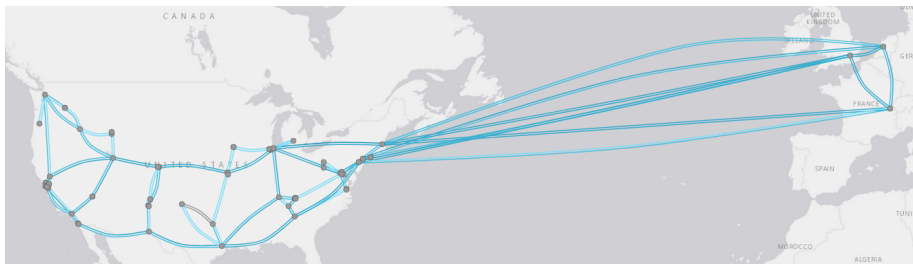


Figure 4: ESNet ²

¹Mogyorósi et al.: Adaptive Protection of Scientific Backbone Networks Using Machine Learning (*IEEE TNSM*)[1]

²source: <https://my.es.net/>

- 3 months examined: 2024-08-01 – 2024-11-30

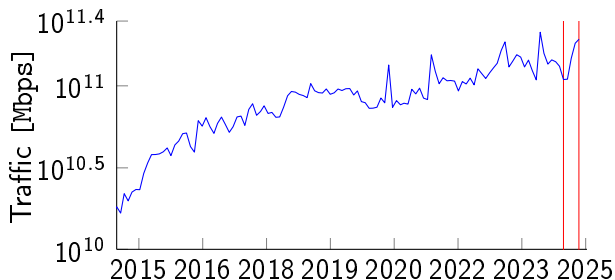


Figure 5: Total bandwidth usage ¹

¹source: <https://my.es.net/>

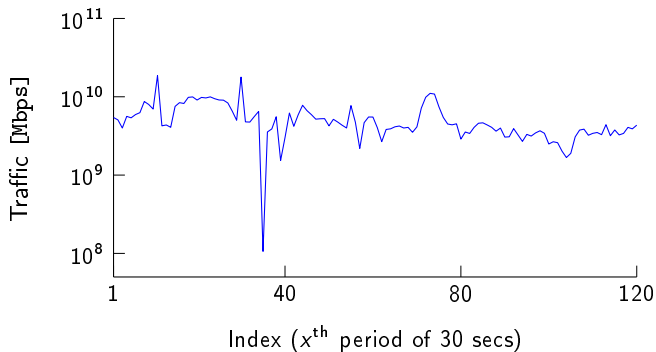


Figure 6: Example: LASV–LOSA 2024-08-01 00:00:00-01:00:00 ¹

¹source: <https://my.es.net/>

Used prediction methods

- ARIMA – Autoregressive Integrated Moving Average
 - less resource-intensive
 - "explainable"
- TCN – Temporal Convolutional Networks
 - may work better on difficult cases

ARIMA – Autoregressive Integrated Moving Average

$$X'_t = c + \sum_{i=1}^p \phi_i X'_{t-i} + \sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t$$

- X'_t – state of the (d -th order differenced) time series at time t
- c – "drift" / "trend"
- ϕ_i – autoregressive term (p of them)
- θ_i – moving average term (q of them)
- ϵ – "white noise" / "random walk" / $\mathcal{N}(0, 1)$
- d – order of differencing ²

²source: Forecasting: principles and practice [2]

ARIMA – Autoregressive Integrated Moving Average

- Determining the optimal ϕ, θ : using Maximum Likelihood Estimation for previous N values
- Recalculate for every time-step t
- Determining the optimal p, d, q : by trial and error...

TCN – Temporal Convolutional Networks

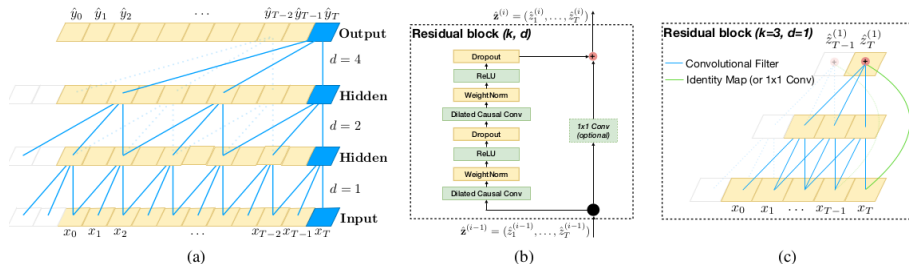


Figure 7: The TCN architecture ³

- Causal convolutions: enforced via right-shifts in subsequent layers
- Residual connections: a highway for gradients

³Bai et al.: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling[3]

TCN – Temporal Convolutional Networks

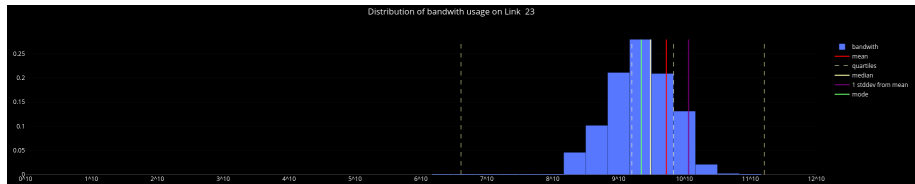


Figure 8: The data distribution is $\sim$ log-normal on most links, when 0-values are removed.

TCN – Temporal Convolutional Networks

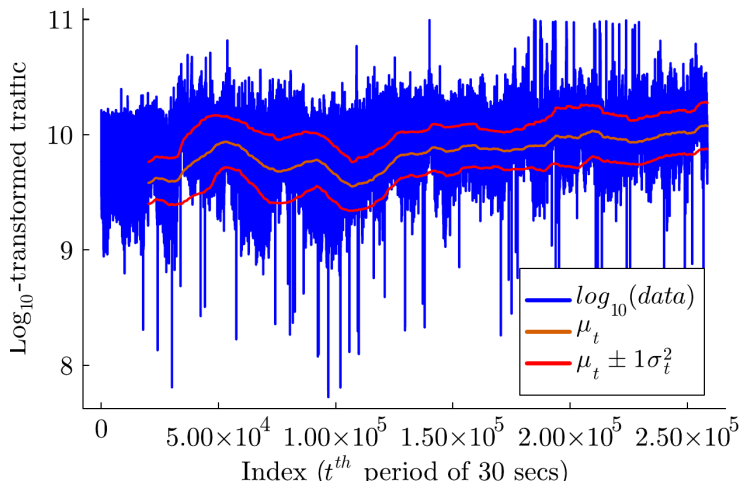


Figure 9: Normalization is crucial: normalize using a running mean and variance.

TCN – Temporal Convolutional Networks

- Train (and validate) on the first 85% of the data
- Test on the rest
- Possible TCN hyperparameters: number of layers, kernel size, number of channels, size of lookback window, dropout rate, etc.

TCN – Unfavorable cases

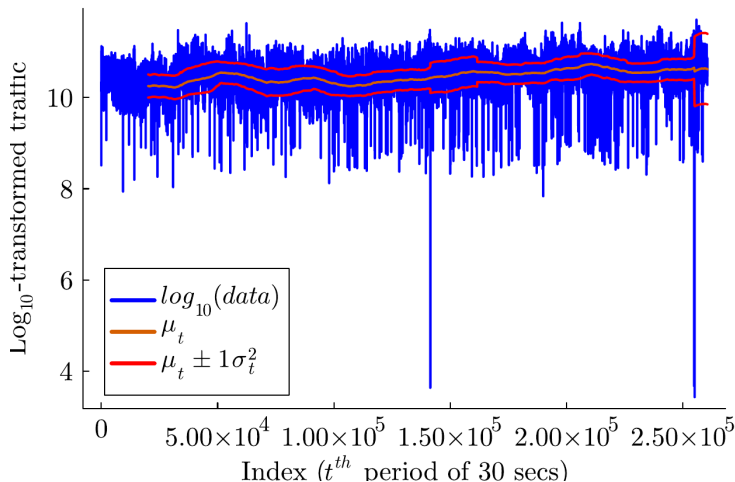


Figure 10: Unfavorable case. Note the sudden increase in variance after $t = 2.5 \times 10^5$

Evaluation results

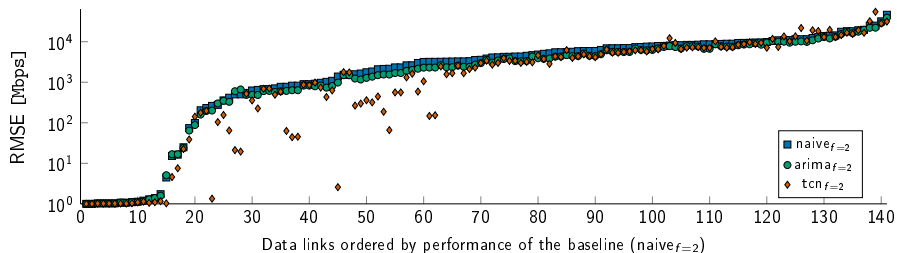


Figure 11: Model performance evaluation on a per-link basis, with a forecast distance of 2 units.

Evaluation results

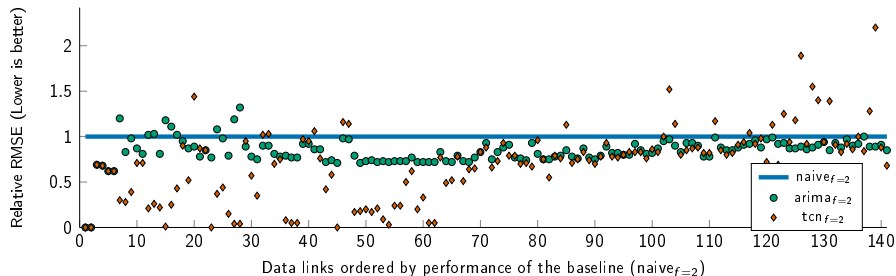


Figure 12: Model performance compared to the baseline on a per-link basis, with a forecast distance of 2 units.

Evaluation results

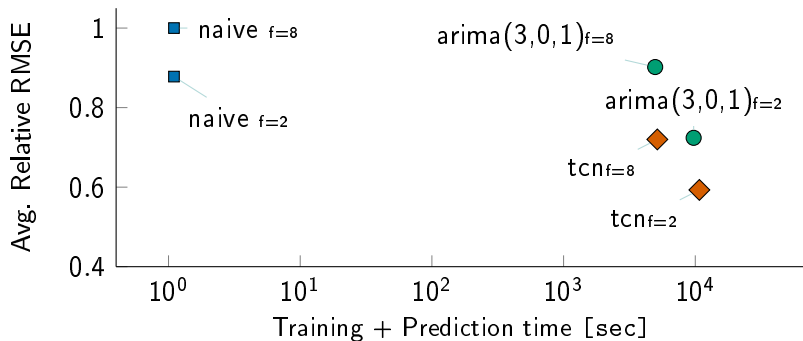


Figure 13: Average of per-link relative model losses vs absolute runtimes

- Conclusions:
 - TCNs work better than ARIMAs in most cases
 - Normalization is crucial for TCN performance
 - Prediction for shorter intervals is (of course) easier
- Future directions:
 - Collect more data.
 - Using multiple input channels with different running window-based transformations (normalizations) of the same data
 - Devise a full adaptive protection scheme

Thank you for Your attention!



Ferenc Mogyorosi, Alija Pasic, Richard Cziva, Peter Revisnyei, Zsolt Kenesi, and Janos Tapolcai.

Adaptive protection of scientific backbone networks using machine learning.

IEEE Transactions on Network and Service Management,
18(1):1064–1076, March 2021.



RJ Hyndman.

Forecasting: principles and practice.

OTexts, 2018.



Shaojie Bai, J. Zico Kolter, and Vladlen Koltun.

An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.

CoRR, abs/1803.01271, 2018.