Multi-Cluster Reconfiguration with Traffic Prediction in Hyper-Flex-LION Architecture

Sandeep Kumar Singh, Roberto Proietti, Che-Yu Liu, and S. J. Ben Yoo

Next Generation Networking and Computing Systems (NGNCS) Group, University of California, Davis, CA, USA

{*sansingh,rproietti,cyliu,sbyoo*}*@ucdavis.edu*

Abstract: We study the performance of Hyper-Flex-LION optical interconnect architecture under dynamic traffic with traffic-prediction-aided multi-cluster reconfiguration. The simulation results show a 17.2% latency improvement and 36.9% packet loss reduction as compared to a fixed topology.

1. Introduction

Reconfigurable flat optical architectures enabled by silicon-photonic switches are emerging as an alternate to today's fat-tree-based high-performance computing (HPC) and data center (DC) network architectures to handle communication-intensive applications with low latency requirement [1, 2]. The key to network reconfiguration is to steer more bandwidth to the hot-spot links. In a dynamic traffic scenario, this requires either to estimate or predict the traffic to adapt the topology as per future demands [3]. Although traffic prediction is a challenging task, DC traffic characteristics have been exploited by machine learning (ML) models in traffic prediction [3–6]. Nevertheless, how to use the predicted traffic to adapt a multi-cluster HPC/DC topology to match future communication patterns is still an open question in DC and HPC networks.

In this work, we propose a traffic-prediction-assisted multi-cluster topology and bandwidth reconfiguration method. We use a long-short term memory (LSTM)-based encoder-decoder recurrent neural network to train time-varying Top-of-Rack (ToR)-to-ToR traffic matrix, and utilize it to reconfigure the wavelengths over fiber links connecting ToRs of reconfigurable Hyper-Flex-LION architecture [2]. Our simulation results show a 17.2% improvement in the end-to-end packet latency and 36.9% improvement in packet loss rate with the reconfiguration scheme when compared to a fixed architecture without reconfiguration.

2. Multi-FSR-based Reconfigurable DC Network Architecture

Fig. 2(a) depicts the architecture, which comprises of three layers: user plane, control and management plane, and data plane. The application manager places the workloads of user plane into the servers and informs a network manager about the new job mapping and its communication requirements. The network manager reconfigures the underlining network topology using a software-defined network (SDN) controller, which calls a routing, bandwidth and topology reconfiguration module to compute topology based on current and predicted traffic to minimize traffic disruption. The SDN controller reconfigures the related ToRs (via OpenFlow by distributing new flow entries) and the optical switch in each cluster. There is a Flex-LIONS optical switch (demonstrated in [7]) interconnecting a group of P ToR switches in each cluster. These clusters are interconnected in rows and columns using multiple Flex-LIONSs to effectively build a Hyper-FleX network (a reconfigurable Hyper-X [2]). Here we leverage two different free spectral ranges (FSRs) of Flex-LIONS [7]. FSR0 guarantees a fixed hierarchical all-to-all connectivity while FSR1 is used to implement the interconnect bandwidth reconfiguration (see Fig. 2(b)). This allows to maintain a shortest path connectivity during and after reconfiguration. The traffic prediction module estimates the traffic demands between the ToRs for next timestep using a neural network model trained with historical data from the database. The reconfiguration is triggered either periodically or intelligently.

3. Traffic Prediction-based Multi-Cluster Network Reconfiguration Scheme

Let $D_t = \{d\}_t^{i,j}, i, j = 1, 2, \ldots, N$ be an $N \times N$ ToR-to-ToR traffic matrix measured at time t. The aim of a traffic prediction (or regression) model is to forecast the subsequent L $N \times N$ traffic instances, given the last T time instances prior. Thus, a regression model estimates $\widehat{\mathbf{D}}_{t+T:t+T+L} = f\left(\mathbf{X}^i_{t:t+T}, \mathbf{W}\right)$, with $\widehat{\mathbf{D}}_{t+T:t+T+L}$ the traffic forecast for L time instances, $\mathbf{W}$ the model weights optimized during the training process and $f(\cdot)$ the estimator *per se*. Among the recurrent networks, the LSTM encoder-decoder is known for its superior performance at temporal prediction and it constitutes the learning algorithm for our work.

Given the predicted traffic matrix $\widehat{\mathbf{D}}_{t+1}$, the number of ports to connect a ToR to other ToRs in a cluster K, and a topology connectivity graph G_0 based on the fiber interconnects, the multi-cluster network reconfiguration

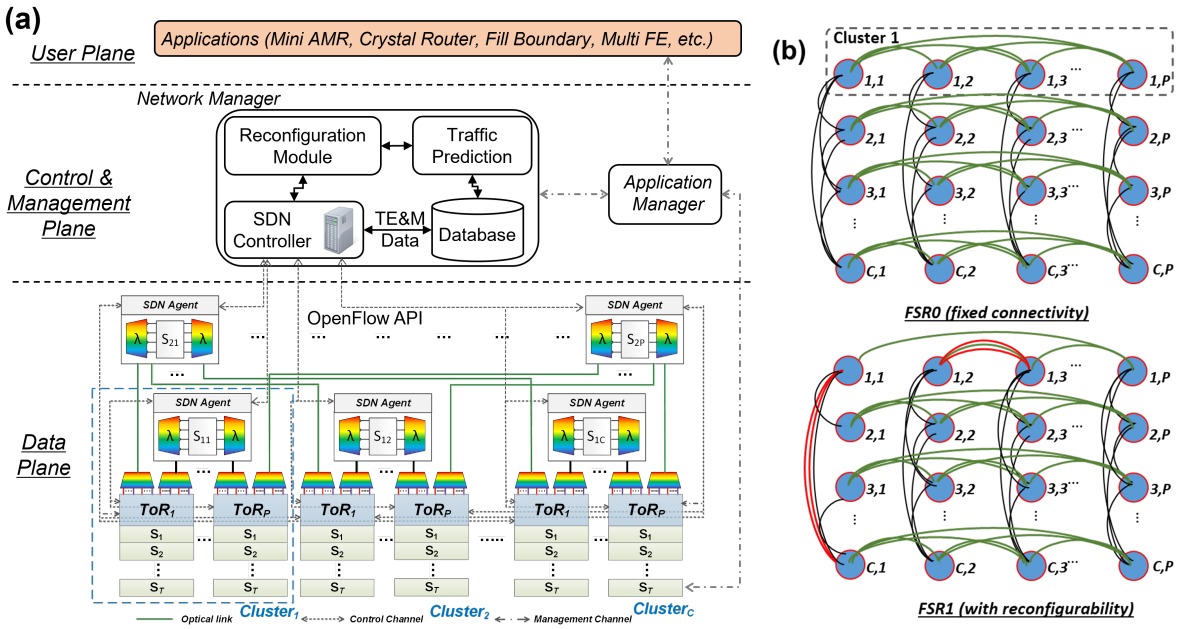

Fig. 1. (a) 2D Hyper-FleX LION interconnect architecture. (b) Clusters are organized into rows with P ToRs per cluster. FSR0 guarantee fixed connectivity. FSR1 enables reconfiguration (see red links).

Algorithm 1 Multi-cluster connectivity graphs computation.

Input: Weight $W_{t+1} \leftarrow$ Normalized predicted traffic $\widehat{\mathbf{D}}_{t+1}$, connectivity graph G_0. **Output**: topology G_{t+1}
$\mathbb{P} \leftarrow$ a set of shortest paths from all-to-all source-destination $(s-d)$ pairs based on a connectivity graph G_0.
while ToRs' port-pairs are available in any cluster, or $\max W_{t+1}$ is not $-\infty$ **do**
    Select a $s-d$ pair (i,j) which maximizes the product of weight vector $w^{i,:}$ and available ports.
    Iterate over each hop on the shortest path $p_{i,j} \in \mathbb{P}$ from node i to j.
    If all hops have available port-pairs, add a link to each hop on $p_{i,j}$, decrease $w^{i,j}$ by a wavelength capacity.
    Otherwise, repeat previous two steps on another shortest path. If not, assign $w^{i,j} \leftarrow -\infty$.
end while
Connect remaining available port-pairs in each cluster of G_0 based on the decreasing order of traffic $\widehat{\mathbf{D}}_{t+1}$.

algorithm recomputes the topology at time t for next time interval. Our proposed multi-cluster topology reconfiguration scheme is summarized in Algorithm 1. The basic idea is to iteratively interconnect ToR ports on the shortest path of larger amounts of traffic to be provisioned over largest number of available ports. We utilize one-step ahead computed topology to identify ports and corresponding links to be added or removed when a reconfiguration is triggered. More importantly, we adopt a *block-before-reconfigure* approach to stop accepting new packets to ports to be reconfigured to reduce packet loss during the reconfiguration phase.

4. Results

We used Netbench simulator to study the performance of reconfiguration in a 2-FSR-based Hyper-FleX-LION under dynamic traffic. The capacity of all links is 10 Gb/s with 20 ns delay, and the port can buffer 150K bytes data. The number of ToRs and servers per ToR is 16, which are organized in a 4 clusters. The servers across multiple clusters generate non-uniform traffic based on four HPC applications traces (i.e. Fill Boundary, Crystal Router, MiniFE and MiniDFT) in a sequential order [8]. The flow arrival events are generated with a Poisson process with rate λ_t. At each arrival event, one source-destination pair is selected from a spatial traffic probability distribution of a running application. Furthermore, to demonstrate the dynamic temporal traffic, we considered a lognormal distribution with mean as a superposed sinusoidal function and variance as 1 to train and predict the test traffic matrices, as shown in Fig. 2. The equal-cost multipath (ECMP) routing with flow splitting mechanism is used to forward the packets over multiple wavelengths of shortest paths. The LSTM encoder-decoder model is trained to predict traffic for next timestep with the window length 4. Each encoder and decoder has 100 LSTM cells, and the model is trained with an Adam optimizer with a learning rate 0.001.

Fig. 2 (left) shows the overall arrival rate of traffic. 70% is used for training, 10% for validation and 20% for testing. The mean square error loss for the normalized training and validation dataset is $\tilde{0}.003$ after 50 epochs. Fig. 2 (right) shows the overall flow arrival rate from ToR 0 to 1 on top, and from ToR 3 to 4 in the bottom. We can see that the model predicts well not only the overall arrival rates, but also individual ToR-to-ToR pairs. Fig.

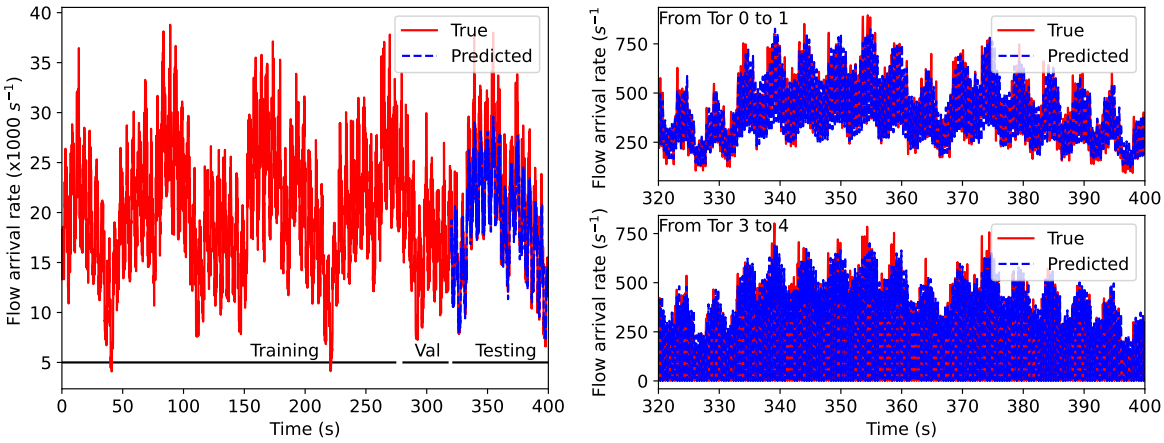

Fig. 2. True and predicted overall traffic flow rate (left), and ToR-to-ToR traffic data (right).

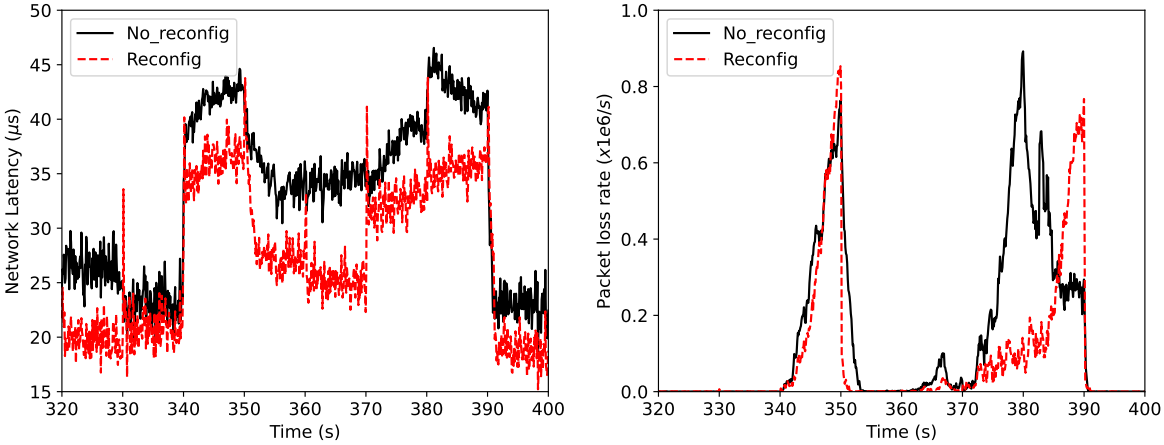

Fig. 3. (Left) Average end-to-end packet latency, (right) packet loss rate w/ and w/o reconfiguration.

3 depicts the effect of predicted traffic to logical topology extraction when we reconfigure every 10 seconds with a reconfiguration time of 100 ms. We also evaluate the algorithm under fixed topology, i.e., no reconfiguration scenario. The left plot shows network latency, i.e., average end-to-end packet latency for the time-varying test data. First, we can see the reconfiguration reduces the packet latency. The percentage improvement in latency by the reconfiguration scheme over fixed topology is 17.2%. Furthermore, the packet latency exhibits abrupt transition (low-to-high) due the reconfiguration process. Fig. 3 (right) shows the packet loss rate under the fixed and reconfiguration scenarios. Interestingly, both exhibit high packet losses when the traffic is either increased from low to high (at $\sim t = 340s$) or remains higher ($t = 370 : 390s$), which is also observed in latency. Nevertheless, the packet loss rate improvement for the reconfiguration scenario as compared to the no reconfiguration is 36.9%.

5. Conclusions

We studied the performance of a ML-assisted multi-cluster reconfiguration scheme in a Hyper-Flex-LION interconnect architecture. The results show 17.2% and 36.9% improvement in packet latency and loss, respectively.

References

1. M. Y. Teh *et al.*, "Flexspander: optical bandwidth steering," *JOCN*, vol. 12, no. 4, pp. B44–B54, 2020.
2. G. Liu *et al.*, "3d-hyper-flex-lion for reconfigurable all-to-all hpc networks," in *SC20*, 2020, pp. 1–16.
3. X. Gao *et al.*, "Experimental assessment of traffic prediction for dcn reconfiguration," *ECOC 2021*.
4. A. Azzouni and G. Pujolle, "Neutm: for traffic matrix prediction in sdn," in *NOMS*. IEEE, 2018, pp. 1–5.
5. S. K. Singh and A. Jukan, "Machine-learning-based prediction," *JOCN*, vol. 10, no. 10, pp. D12–D28, 2018.
6. M. Balanici and S. Pachnicke, "Machine learning-based traffic prediction," in *OFC*. OSA, 2019, pp. Th1H–4.
7. X. Xiao *et al.*, "Multi-fsr silicon photonic flex-lions," *JLT*, vol. 38, no. 12, pp. 3200–3208, 2020.
8. NERSC, "Characterization of the doe mini-apps. https://portal.nersc.gov/project/CAL/doe-miniapps.htm."

This work was supported in part by ARO award # W911NF1910470, DoD award # H98230-19-C-0209, NSF ECCS award # 1611560, and by DoE UAI consortium award # DE-SC0019582, DE-SC0019526, and DE-SC001969.