Degree-based Balanced Clustering for Large-Scale Software Defined Networks

Talha Ibn Aziz[1] Shadman Protik[1] Md Sakhawat Hossen[1]
Salimur Choudhury[2] Muhammad Mahbub Alam[1]

[1]Department of Computer Science and Engineering
Islamic University of Techology
Board Bazar, Gazipur, Bangladesh

[2]Department of Computer Science
Lakehead University
Thunder Bay, Ontario, Canada

IEEE Wireless Communications and Networking Conference,
15-19 April, Marrakech Morocco

Table of Contents

Software Defined Networks (SDNs)

- Software Defined Networks (SDN) decouples the network layer of the traditional protocol stack into the control plane and the data plane
- The control plane controls the path selection mechanism and the data plane follows the mechanism to forward data packets.
- Devices with high processing capacity called ‘controllers’ communicate with the data plane devices (switches) as per requirement.
- The initial design of SDNs included a single controller in a network.

Controller Placement Problem (CPP)

- The single controller design gave rise to issues like scalability, reliability - a single point of failure, security, and the formation of a bottle-neck due to overflow of switch-controller queries.
- Researches propose a multiple controller approach which introduced a few questions-
  1. How many controllers?
  2. Where to place them?
  3. Which switch falls under which controller?
- Several parameters need to be optimized to place multiple controllers, making the solution to the Controller Placement Problem (CPP) NP-Hard.

Motivation

- To the best of our knowledge,
  - No optimal solution to the CPP has yet been proposed.
  - The researches done are either exhaustive or work with optimizing a single parameter (latency, reliability, etc.).
- Our proposed algorithm Degree-based Balanced Clustering places controllers in polynomial time complexity. DBC simultaneously deals with multiple parameters.
- DBC outperforms state-of-the-art controller placement algorithms like Density Based Controller Placement (DBCP), in terms of Flow-Setup Latencies and Load Balancing.

Contribution

DBC is a novel and efficient controller placement algorithm which optimizes the following parameters:

- **Flow setup latency** - The maximum latency incurred to set the path for a new data packet.
- **Route synchronization latency** - Synchronization delay in case of a change in the network due to a link or switch failure.
- **Load of a controller** - The balancing of loads handled by the controllers. The switches are assumed to have identical load.

In order to achieve the mentioned goals, DBC forms balanced clusters in terms of both cluster size and inter-switch distances.

Degree-based Balanced Clustering

Our proposed algorithm performs Controller Placement in three phases.

1. **Cluster Formation** - Group switches for placing a controller (in each group).
2. **Controller Selection** - Select one controller among the switches in a group based on some parameters.
3. **Selection of optimum number of controllers** - Change the number of controllers and repeat the process to find the optimal configuration.

Cluster Formation

1. Find an approximate cluster radius (hop count).
   - Start with a cluster containing only one switch and add switches to the cluster based on average degree.
   - In each iteration, increment cluster radius by one and add an expected number of nodes to the cluster.
   - Stop incrementing when the number of nodes exceeds the approximate switch count for a balanced cluster.
2. Select the switch with highest connectivity as cluster head. We assume that the maximum node degree corresponds to the highest connectivity.
3. Assign each switch to the nearest cluster head.

Controller Selection

- We select controllers based on weighted-sum of inter-cluster and intra-cluster distances.
- Intra-cluster distance for a switch is the average of the distances from the switches of the same cluster.
- Inter-cluster distance for a switch is the average of the distances from the switches of the other clusters.
- We select the switch with the least weighted-sum in a cluster as the controller of that cluster. Weights can be used to control the extent of inter-cluster or intra-cluster influence while selecting a controller.

Optimal number of controller

- Calculate the flow-setup latency for the current network configuration.
- Increment the number of controllers and repeat steps 1 (Clustering) and 2 (Controller Selection). Calculate the new route-synchronization latency.
  - If the improvement percentage is below a certain threshold, terminate the process.
  - Otherwise, continue in the same way and increment the number of controllers for another improvement calculation.

Experimental Analysis

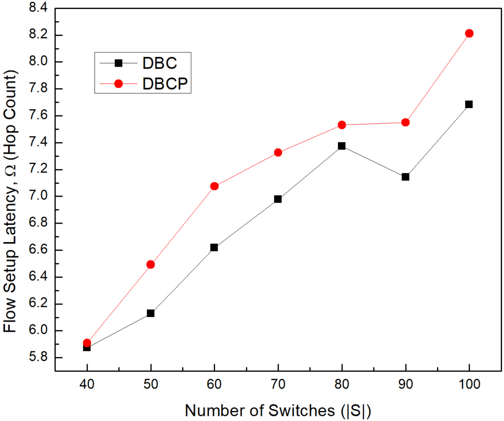

Figure: Comparison between DBC and DBCP in terms of Flow Setup Latency for identical numbers of controllers

Experimental Analysis

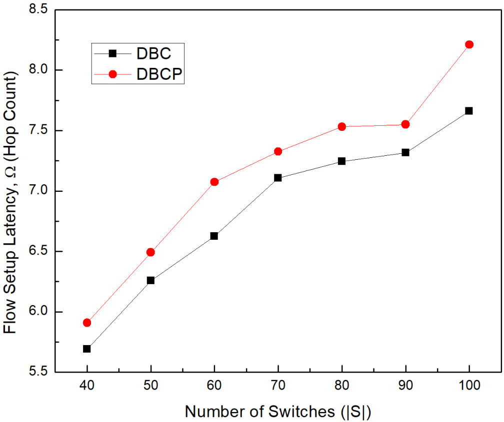

Figure: Comparison between DBC and DBCP in terms of Flow Setup Latency for different numbers of controllers.

Experimental Analysis

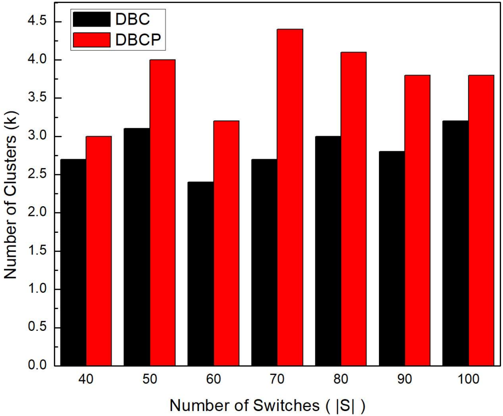

Figure: Comparison between DBC and DBCP in terms of Number of Controllers.

Experimental Analysis

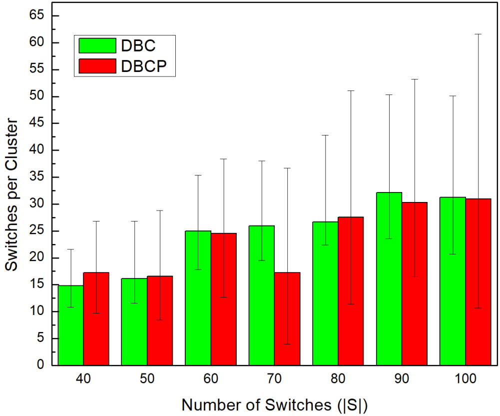

Figure: Comparison between DBC and DBCP in terms of maximum, minimum, and average switch distribution.

Conclusion

- DBC outperforms DBCP in terms of flow-setup and route-synchronization latencies.
- DBC balances the load of the controllers.
- Future work may include
  - Working with graph representations of networks having variable edge weights (for example - link latencies).
  - Considering variable controller-processing powers and dynamic switch loads.