**CONTROLLER PLACEMENT PROBLEM**

In SDN control plane is separated from data plane. Control plane consists of controllers. Data plane consists of switches which merely does the packet forwarding. In earlier days a single controller was used. But a single controller is not enough for large networks. It will affect the performance and scalability of the network. So multiple controllers were introduced. Example for distributed implementation of the controllers are Onix ,Devoflow, Kandoo, and HyperFlow. However deploying a set of controllers in a large network is a big challenge. An important issue in SDN is placement of controllers. Controller placement problem(CPP) was first identified by B Heller el al[1] in 2012. Controller placement problem can be defined as a design choice of the following two questions.

(1) How many controllers are needed?

(2) Where in the topology should they go?

In [1], the authors motivated the controller placement problem and the placement of controllers based on average and worst-case propagation latency between switch and its assigned controller was done. When a packet enters a switch, the decision where to forward the packet is taken by controller by looking on the flow table. So the propagation latency is an important metric for determining the performance of the network. Their aim was to minimize the latency between switch and its assigned controller. For average propagation latency, K- median algorithm is used for optimization. For worst case propagation latency, K- center algorithm is used for optimization. This optimization is equivalent to facility location problem which is NP hard.

**On the Capacitated Controller Placement Problem in Software Defined Networks**

In [2], Yao et al. improved Heller's work by considering load of controllers along with latency. This new problem is called capacitated controller placement problem (CCPP). Load is a critical factor in the placement of controllers due to the following reasons.

1. The server capacity limitation: The server on which the controller is placed may have a certain memory capacity and access bandwidth. It cannot manage so many switches due to its capacity limitation. So load is a critical factor for controller placement.
2. The latency of message processing: For heavy loaded controllers, the latency of message passing will increase. Whenever the load of a controller reaches a threshold, the message processing latency on the controller will increase. In such cases, the controller processing latency will turn out to be a non-negligible factor in the total round-trip latency.
3. Failure: Heavy-loaded controllers have higher failure probability, because they have fewer resources to handle various errors and are more likely to be attacked. In some cases, the failure of heavy-load controller may even cause cascading failures of other controllers.

The placement of controllers should be done in such a way that it should minimize the propagation latency between switch and its assigned controller, whereas the load of each controller should not exceed its capacity. Since we are considering load along with latency, here the Controller Placement Problem (CPP) is known as the Capacitated Controller Placement Problem (CCPP). B. Heller et al. used K-center algorithm. To solve capacitated controller placement, capacitated K-center algorithm is used. This is the first paper which has considered the load on controllers. The network can be represented as a graph. CCPP can be represented using mathematical model. CCPP is a NP hard problem.

The load of a controller is mainly formed by

1. Processing the PACKET\_IN events and delivering the events to the applications;
2. Maintaining the view of the local network partition;
3. Communicating with other controllers to form a global view;
4. Installing the flow entries generated by applications.

The most significant part of the total load is the load of processing PACKET\_IN events. So the arriving rate of PACKET\_IN events on a controller is used as the measurement of load on controller.

To find the minimal number of controllers with a specified radius r, an Integer Programming model SP(r), proposed in [9] is used. SPLR(r) is the linear relaxation of SP(r). This algorithm has two phases: in phase I, the lower bound of radius is obtained by solving SPLR in binary search; in phase II, the radius is increased from lower bound until a placement is found. Yao et al modified the algorithm by modifying the following things in phase I. binary search only considers the distance between any pair of locations, rather than all integer numbers in a given range. This modification ensures faster convergence. When r is small, it will cost a lot of time to solve SPLR(r). However, even measured in kilometres, the distances between each pair of locations are typically large numbers. Because the binary search in integer numbers converges until the step is less than 1, it requires more iterations than searching in possible distances. Because the possible radius must be one of the distances, searching in possible distances will not omit the result radius.

**Pareto-optimal resilient controller placement in SDN-based core networks**

Hock et al. improved the controller placement by considering resiliency metrics al which are important to SDN network along with propagation latency between switch and controllers. The controller placement problem can be considered as a multi objective optimization problem. By studying several performance and resilience metrics, Hock et al.[4] proposed a novel controller placement in SDN which improve resiliency and failure tolerance. When we consider resiliency metrics and latency, no single best controller placement is available; we have to find a balanced trade-off between these metrics. So Hock et al. used pareto optimization. Based on this pareto optimal controller placement, they developed a matlab based framework based called POCO In this framework, an exhaustive evaluation of all possible placements has been performed for the network topology.

POCO computes resilient pareto-based optimal controller placements. The advantage of this approach compared to any particular mixed integer linear program or heuristic is that it evaluates the entire solution space and provides information about the objectives considered for placement. Regarding multi criteria or multi objective optimization that means that no decision has to be taken before invoking the optimization by defining some constraints or weighted objective functions. In contrast, offering all possible solutions evaluated by all objectives, offers the possibility to take the decision afterwards.

**Resilient Controller Placement for SD**N

A main objective for a good controller placement is to minimize the latencies between nodes and controllers in the network. However, looking only at delays is not sufficient. A controller placement should also fulfil certain resilience constraints. In the following, we briefly explain these issues and what is necessary to be resilient against them.

1) Controller Failures: As illustrated by Heller et al., a larger number of well-distributed controllers in a network can help to lower the maximum latency between the nodes and their controllers. It also increases the failure tolerance if some of the controllers stop working. Zhang et al. assume in their work that a node is not able to route any more if it loses its connection to the controller. However, we suppose that in case of a controller outage, it is possible to reassign all nodes previously attached to that controller to their second closest controllers in the network using a backup assignment or signaling based on normal shortest path routing. Thus, as long as at least one of the controllers is still reachable, all nodes keep being functional. However, the latencies of the reassignednodes to their new controller can be significantly higher than the latencies to the primary controller.

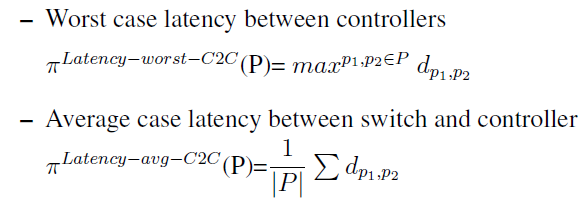
2) Network Disruption: In contrast to controller failures, the outage of network components, such as links and nodes, often has a much higher impact on the network stability, as it alters the topology itself. The shortest paths between some of the nodes change, leading to different latencies and possibly to the reassignment of nodes to other controllers. Even more severe is that entire parts of the network are in danger of being cut off by link or node outages. In the worst case, some nodes can no longer be connected to a controller as they are cut off from all controllers. These nodes are still working and able to forward traffic, but cannot request instructions anymore. We have to assign new controllers or link between switch and controller to address any new functionality realized by controller.

3) Load Imbalance: we assume that nodes are always assigned to their nearest controller using latency as metric, i.e. the shortest path d(n; c) between the node n and controller c. The more nodes a controller has to control, the higher is the load on that controller. This is especially relevant in scenarios where nodes communicate often with their controller, e.g. when considering NFV. If the number of node-to-controller requests in the network increases, so does the chance of additional delays due to queuing at the controller system. To be resilient against controller overload, the assignment of nodes to the different controllers should be well-balanced. Controller performance can vary among connected switches. This stresses the importance of an intelligent controller placement that also takes load balancing aspects into account.

Depending on the use case, it can be desirable to have roughly equal load on all controllers, so that no controller is overloaded while others have only little work to do. In the following, we address a good balance of the node-to-controller distribution. As formal metric, we introduce the balance of a placement or rather the imbalance, imbalance, i.e. the offset to a totally balanced distribution, as the difference between the number of nodes assigned to the controller with the most nodes and the number of nodes assigned to the controller with the fewest nodes.

Where P is the possible controller placements

4) Inter-Controller Latency: It is clear that a single controller is not enough to reach any kind of resilience in a network. However, when several controllers are placed in the network, another issue arises. If the control logic of the network is distributed over several controllers, these controllers need to synchronize to maintain a consistent global state. Depending on the frequency of the inter-controller synchronization, the latency between the individual controllers plays an important role. For the depicted placement, the messages between the controllers have to travel relatively long distances in the network, which might not be acceptable.



**Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks**

Pareto optimal controller placement performs an exhaustive evaluation of all possible placements based on different placement metrics. It can be performed within a practically feasible time frame for small and medium-sized networks, such an approach is out of scope for large problem instances which have significantly higher time and memory requirements. To decrease the computation time and memory, heuristics methods are used.

In the context of finding the global optimum of a function that has a large domain, i.e., the optimization problem has a large search space; simulated annealing is a popular heuristic approach. Simulated annealing is a Monte Carlo method and has two distinctive properties. First, during the exploration of the search space, moves to solutions worse than the current one are permitted in order to avoid getting stuck in a local optimum. This is achieved by incorporating a control parameter that is referred to as temperature, which determines the probability of accepting such moves. Second, the probability of moving to a worse solution gradually decreases with the number of iterations. Additionally, the acceptance probability depends on the difference between the objective values of the current and the proposed solution. The rationale behind this behaviour is that accepting rather bad solutions at the beginning allows for a broader coverage of the search space while the lower acceptance probability at the end helps with convergence.

**Controller placement in software-defined WAN using multi objective genetic algorithm**

Evolutionary algorithms and genetic algorithms are capable of performing multi objective optimization. In [5], the authors addressed the controller placement problem with respect to latency between switches and its assigned controllers, among controllers themselves and load balancing. The NSGA –II (second version of Non dominated Sorting Genetic Algorithm) was adapted which proved to be efficient for obtaining a diverse approximation set of the pareto optimal front considering these objectives.

**A K-means-based network partition algorithm for controller placement in software defined network**

Wang et al.[7] proposed a new approach with network partition technique to solve the controller placement problem. By modelling the network partition problem, they proposed an effective network partition algorithm named optimized K-means to partition network into k sub networks in terms of latency. The network partition technique is introduced to simplify the controller placement problem. The problem is not NP hard.

**The SDN Controller Placement Problem for WAN**

Large networks are always partitioned into several small networks when deploying software defined networking. The network partitioning with SDN control planes open many unanswered questions such as latency, reliability, and load balancing. This paper opens the investigation by focusing on two specific questions: given a wide-area network topology, how to partition it into several small SDN domains, and where should the controller go in each SDN domain? To answer these questions, they proposed a novel approach to efficiently and accurately evaluate SDN controller placement problem for WAN. This approach uses the Spectral Clustering placement algorithm to partition a large network into several small SDN domains.

**COMPARISON**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Name | Metric considered | How | Year |
| 1 | The Controller Placement Problem | Average-case latency  Worst-case latency | minimum k-median problem  minimum k-center | 2012  IEEE Conference |
| 2 | On the Capacitated Controller Placement Problem in Software Defined Networks | Propagation latency and load on computers | Capacitated k-center problem | 2014  IEEE letter |
| 3 | POCO-Framework for Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks | Resiliency Metrics:  Controller Failures, Network Disruption, Load Imbalance, Inter-Controller Latency, Latency between switch and controller | Pareto Optimal placement | 2013  IEEE conference |
| 4 | Heuristic Approaches to the Controller Placement Problem in Large Scale SDN networks | Latency, Load balancing | Pareto Simulated Annealing algorithm | 2015  IEEE  transaction |
| 5 | The SDN Controller Placement Problem for WAN using multi objective genetic algorithm | Latency, Load balancing, | NSGA-II algorithm | 2015  IEEE  Conference |
| 6 | A K-means-based Network Partition Algorithm for Controller Placement in Software Defined Network | Latency | K-means clustering | 2016 IEEE Conference |
| 7 | The SDN Controller Placement Problem for WAN | Latency | Spectral Clustering | 2015 IEEE Conference |

**OBSERVATIONS**

* K- center approach can be used only when latency is considered. It can not be used for multi objective controller placement.
* Pareto optimization gives an optimal multi objective controller placement. But for large networks exhaustive evaluation is needed and it takes more computational time and memory
* So heuristic methods are used for multi objective controller placement. Even though they give solution faster and the need less memory, the solution not accurate. It gives an approximate solution.
* To simplify controller placement, network partition can be used. K- Means clustering can be used for network partition.
* But none of the above approaches give a proper controller placement in the perspective of load balancing, scalability and optimal number of controllers.
* All previous methods need to specify the number of controllers as input.

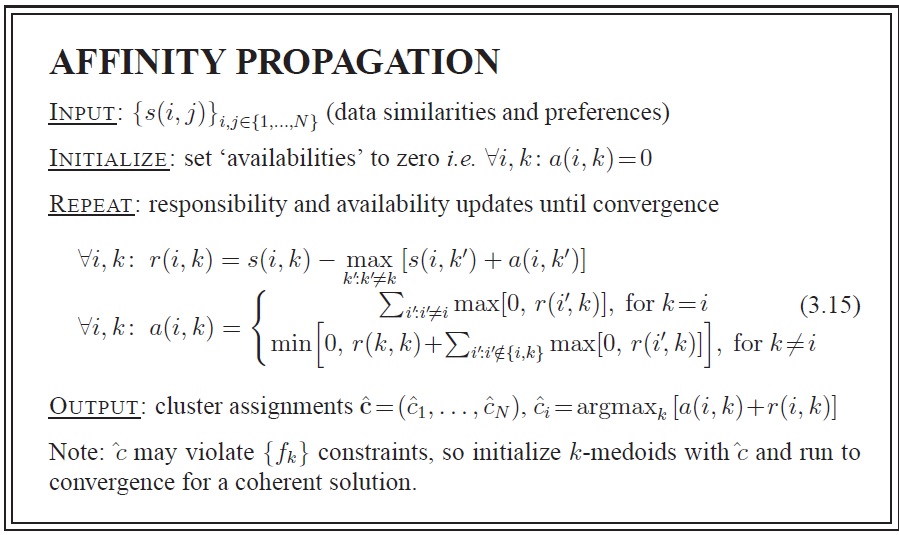
**PROPOSED SYSTEM**

Our aim is to determine the number of controllers and their location in software defined networking using affinity propagation clustering.It is a clustering based controller placement. Partition the network into sub networks using affinity propagation algorithm. Affinity propagation(AP) clustering automatically determines the number of sub networks. AP is an exemplar based message passing clustering method. An exemplar is a data point that is representative of the cluster. The objective of affinity peopagation is to identify a subset of data points as exemplars and assign every other data point to one of those exemplars. Input is the set of real-valued pairwise similarities , s(i,k) between nodes(data points). Two kinds of messages are passed between data points

1)Responsibility: Messages sent from cluster members (data points) to candidate exemplars (data points), indicating how well-suited the data point would be as a member of the candidate exemplar's cluster.

2) Availability:Messages sent from candidate exemplars (data points) to potential cluster members (data points), indicating how appropriate that candidate would be as an exemplar.

Affinity propagation takes as input a set of pairwise similarities between data points and finds clusters on the basis of maximizing the total similarity between data points and their exemplars. In our case we will represent the network as a graph. Similarity function will be negative distance between points. On exemplars controllers are placed.



**REFERENCES**

1. B. Heller, R. Sherwood, and N. McKeown, “The controller placement problem,” in Proc. HotSDN , 2012, pp. 7–12.
2. G. Yao, J. Bi, Y. Li and L. Guo, "On the Capacitated Controller Placement Problem in Software Defined Networks," in IEEE Communications Letters, vol. 18, no. 8, pp. 1339-1342, Aug. 2014.
3. Y. Hu, W. Wendong, X. Gong, X. Que and C. Shiduan, "Reliability-aware controller placement for Software-Defined Networks," 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, 2013, pp. 672-675.
4. D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," Teletraffic Congress (ITC), 2013 25th International, Shanghai, 2013, pp. 1-9.
5. A. Jalili, V. Ahmadi, M. Keshtgari and M. Kazemi, "Controller placement in software-defined WAN using multi objective genetic algorithm," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015, pp. 656-662.
6. S. Lange et al., "Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks," in IEEE Transactions on Network and Service Management, vol. 12, no. 1, pp. 4-17, March 2015.
7. G. Wang, Y. Zhao, J. Huang, Q. Duan and J. Li, "A K-means-based network partition algorithm for controller placement in software defined network," 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-6.
8. Peng Xiao, Wenyu Qu, Heng Qi , Zhiyang Li, Yujie Xu: The SDN Controller Placement Problem for WAN; IEEE; 2015
9. F. A. Ozsoy andM. C. Pinar, “An exact algorithm for the capacitated vertex k-center problem,” Comput. Oper. Res., vol. 33, no. 5, pp. 1420–1436, May 2006.