SHORT TERM SCIENTIFIC MISSION (STSM)
SCIENTIFIC REPORT

This report is submitted for approval by the STSM applicant to the STSM coordinator

Action number: 15127
STSM title: Selecting the SDN Controllers Placement to Enhance the Network Robustness to Multiple Node Failures
STSM start and end date: 08/10/2017 to 14/10/2017
Grantee name: Dorabella Martins da Silva Santos

PURPOSE OF THE STSM:

The purpose of this STSM was to start a collaboration among Carmen Mas Machuca, Dorabella Santos and Amaro de Sousa, to investigate SDN controller placement robust to malicious node attacks.
Different controllers are required for a scalable SDN network with acceptable control plane performance (guaranteeing maximum switch-controller delay requirements), avoiding the single point of failure issue. So that the network works as a logically centralized unit, there is also a maximum controller-controller delay requirement.
When malicious attacks occur in the network, shutting down all sites with controllers causes a total network collapse. Having more controllers in the network, can turn the network more robust to these multiple node attacks. Therefore, the aim is to place different controllers so that the network is as robust as possible to malicious node shutdowns, while guaranteeing acceptable control plane performance (at least in the fully operating state, i.e., without failures). The robust controller placement guarantees that if all but one controller places are attacked, the data plane can still guarantee connectivity between the surviving switches and the surviving controller.

DESCRIPTION OF WORK CARRIED OUT DURING THE STSM

During the STSM visit, the work carried out on this robust controller placement problem (CPP), resulted in a conference paper submitted to DRCN 2018.
The robust CPP studied is as follows. For a given maximum switch-controller (SC) delay and a given maximum controller-controller (CC) delay in the regular (i.e., working) state, we aim to find a CPP solution that maximizes the network robustness for a given number of malicious node attacks. First, we guarantee that if all but one controller nodes are shutdown, there is still a switching path from any switch to the surviving controller. We developed an ILP based method aiming to enumerate all such solutions. Then, for different malicious node attacks corresponding to different attacker's strategies, we evaluated the previous solutions to determine the ones that maximize the minimum number of switches that can still be connected to at least one controller (regarding or ignoring the maximum SC delay). We compared the robust CPP solutions with non-robust CPP solutions, the latter aiming solely to minimize the average SC and/or CC delays. For the non-robust CPP, we implemented ILP models that can be efficiently solved by standard solvers.
The computational results (submitted for DRCN 2018) show the trade-off between the robustness improvement of the proposed robust CPP solution against the resulting penalties on the SC and CC delays.

DESCRIPTION OF THE MAIN RESULTS OBTAINED

Consider a SDN switching network, where the delays between two nodes are computed in terms of the shortest path length between them. We consider the non-robust CPP and the robust CPP.

In the non-robust CPP, the aim is to optimize the SDN control plane performance in its regular state, by minimizing the average SC and CC delays, guaranteeing that:

(1) A set of C controllers are placed in the network;
(2) Each switch is connected to a primary controller;
(3) The SC delay between each switch and its primary controller cannot exceed a given D_{sc};
(4) The CC delay between any pair of controllers cannot exceed a given D_{cc}.

To jointly minimize the average SC and CC delays, results in a bi-objective optimization problem with multiple optimal solutions resulting in tradeoffs between the two objectives. In this STSM, we focused on the two extreme tradeoffs:

(a) *MinAvgSC* – the non-robust CPP that minimizes the average SC delay, and then imposing the minimum obtained SC delay, minimizes the CC delay;
(b) *MinAvgCC* – the non-robust CPP that minimizes the average CC delay, and then imposing the minimum obtained CC delay, minimizes the SC delay.

In the robust CPP, the aim is to optimize robustness in the failure state. We assume that each switch can dynamically reconnect to the closest surviving controller, in case its primary controller fails. This means that any controller can be a backup controller for any switch. We consider that when a node hosting a controller is shutdown, both the controller and its collocated switch fail.

Besides conditions (1)-(4), the robust CPP must guarantee an additional condition:

(5) There must be a routing path from each switch to each controller that does not pass through any other controller, i.e., if any $C-1$ controllers are shutdown, the surviving switches can still connect to the surviving controller, maintaining the surviving network fully functional.

Note that in case of failure, conditions (3) and (4) might not be ensured.

In this problem, the aim is to maximize robustness for a given set of malicious attacks. Instead of solving the problem for a given optimization objective of robustness, we chose to obtain all feasible solutions for given C, D_{sc} and D_{cc} sets of values satisfying conditions (1)-(5). In this way, we can then evaluate the solutions against different objectives (or combinations of objectives).

We implemented a basic method for enumerating all solutions by iteratively solving an ILP model. For each obtained solution, a condition eliminating the solution from the feasible set is included for each iteration of the ILP model. If the set of feasible solutions is very large, then the enumeration method can take too long. Therefore, we consider a parameter $L_{\max}$ for the maximum number of solutions, and so the enumeration process ends also when $L_{\max}$ solutions have been obtained.

To accelerate the enumeration process, we have devised a speedup technique where most of the solutions are obtained by an efficient heuristic algorithm. The heuristic algorithm is based on a random walk that starts with a feasible solution and repeatedly jumps from a feasible solution to a neighbor solution by randomly changing the location of one controller. If the neighbor solution is feasible, the random walk progresses with it; otherwise, the procedure progresses from the previous feasible solution. All feasible solutions are stored and the random walk stops when $I_{\max}$ consecutive neighbor solutions are infeasible ($I_{\max}$ is an input parameter).

Once the feasible solutions are obtained, the aim is to select those that maximize the network robustness to a given set of malicious attacks targeting to simultaneously shutdown p network nodes. We consider a set M of such malicious attacks, i.e., each element of M is a unique combination of p network nodes to be simultaneously shutdown.

To define a proper set of malicious attacks, we have assumed, as in other works, that the attacker has knowledge of the network topology but no knowledge of the number of controllers or their location. In this way, it is most probable that the attacker targets nodes that exhibit high centrality measures, hoping to maximize the network disruption. We have considered three centrality based attacks: based on node degree; based on closeness centrality; and based on betweenness centrality. For each centrality measure, to select the p target nodes, the node with highest centrality is chosen. Then, it is eliminated from the graph and the centrality measures are recomputed. The node with highest centrality in the new graph is

selected and eliminated from the new graph. This process continues until p nodes have been selected.

In the computational results, we have used two topologies: Germany50 with 50 nodes, 88 links and average node degree of 3.52 (http://sndlib.zib.de); and CORONET CONUS with 75 nodes, 99 links and average node degree of 2.64 (http://www.monarchna.com/topology.html). We have defined the delays in terms of shortest path lengths. As in other works, the maximum delays D_{sc} and D_{cc} are given as percentages of the graph diameter (the largest shortest path length between any pair of nodes in the network). We have considered $C = p + 1$ to make the network robust to the shutdown of p nodes targeted by centrality-based attacks. The sets of values chosen for D_{sc} and D_{cc}, represent different compromises between the SC and CC delays which are tight but guarantee that both the non-robust and robust CPPs are feasible.

We solved the non-robust CPPs *MinAvgSC* and *MinAvgCC* for all instances. Most of these solutions exhibit the robustness property of there being a routing path from each switch node to each controller node that does not pass through any other controller, even though it was not imposed. We also solved the robust CPP using the speedup technique. The computational results show that the random walk heuristic computes a very large percentage of solutions, while spending a small percentage of the total runtime, for almost all the instances, showing that the speedup technique is very efficient.

Comparing the robust CPPs with the non-robust CPPs, the main conclusions are that the robustness gains become more significant, on average, for sparser networks (i.e., with lower average node degrees) and for attacks shutting down more nodes (although, they vary a lot between different instances). On the other hand, the average SC delay penalties are always small and the average CC delay penalties vary significantly between different instances but, in some of them, can be significant.

FUTURE COLLABORATIONS (if applicable)

The DRCN 2018 paper was a first joint effort on the SDN problem featuring the robustness property presented above. A more comprehensive study of the robustness against other types of attacks must also be conducted. Moreover, other approaches to the problem have also been discussed and are under study. In the near future, we expect to submit a paper to a journal.