Evento	Salão UFRGS 2018: SIC - XXX SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2018
Local	Campus do Vale - UFRGS
Título	Meta-Data Management on Programmable Data Planes
Autor	LUCAS BARBOSA CASTANHEIRA
Orientador	ALBERTO EGON SCHAEFFER FILHO

Universidade Federal do Rio Grande do Sul

Meta-Data Management on Programmable Data Planes

Orientado:Lucas Castanheira Orientador: Alberto Schaeffer-Filho

Programmability has been extensively investigated to enable a more flexible operation of computer networks. Programming a network can be achieved by either modifying its behavior in the control plane, through technologies like OpenFlow, or in the data plane, with technologies like P4. Research efforts have begun to look into data plane programmability in order to break the ossification (i.e. the difficulty for change and deploying new technologies) ensuing from years of legacy hardware. One of the many technologies in this endeavor is the P4 programming language. P4 is a protocol independent language used to modify Software Defined Networks (SDN) switches to very specific technological needs which could not be accommodated by designers into a so called general purpose switch. Because it deals with software and not hardware it also drastically reduces capital expenditures and operational expenditures for the deployment of these novel technologies.

However the programmability offered by P4 is in many ways restricted. Sometimes we will need a more global view of the network to make run time decisions in the data plane. The key point in doing so is determining how to gather, store and distribute the relevant knowledge inside a network's data-plane. It is in our interest to minimize distribution and gathering costs of meta-data to other P4 programs running on the data plane.

Our work is focused on this information management. The proposed approach consists of (i) separating the network into clusters (neighborhoods of forwarding devices that can distribute information among themselves). This is achieved by representing the topology as a graph in which the weight of the edges connecting two switches is a metric which will define how close the nodes are to each other. If we use latency as the metric, our clusters will all be formed of devices that have low latency amongst themselves. Then, (ii) a monitoring system, built to make stateful data promptly available from inside the data plane and if need be, raise alerts about anomalies. This is important because the data plane is a very rich source of information about a network's health. Today's networks, for the sake of performance and simplicity, only keep a small but stateful part of the inner workings of the data plane (e.g. per rule hit counters). Our system tries to leverage programmability by monitoring the data plane directly, which enables us to ascertain which information is worth keeping. Finally, we have (iii) a controller-based runtime data management scheme, which we achieve through the insertion of strategic protocol specific packets we call Crawler Packets (CP), which can distribute and gather information at the same time inside clusters. A Crawler Packet goes through an entire cluster through static forwarding in a cyclic manner that is predetermined but still mutable.

The following scenario can help better understand our strategy. Assume that we want to gather information from every switch in our network. Traditionally in an SDN environment we would go about this problem by using the controller and its direct link to the switches to poll them for the data. This can result in a wide range of connections being opened and stress added to the controller (specially if TCP is used and the network is large). If, however, we were to use our system, we would insert a CP into every cluster and wait for them to return. If TCP was used in the original scenario we would have even more overhead in resources allocated for the many switches and their connections. Our approach allows us to create a metadata management system that is efficient (as the number of messages exchanged between the control and data plane is reduced) and fault tolerant (because if a link goes offline we can change the route of the crawler packet from the controller). And all of these features completely handled by the data plane with minimal effort from the control plane (the insertion of the CP which is the first and only control-plane interaction necessary to start the process of collection/distribution) in contrast to the considerable overhead a controller-based approach would have caused (high bandwidth usage, TCP connections). Future work involves reacting to stateful events we discover from the information maintained by our monitoring system.