Enabling event-based hierarchical synchronization in SDN ONOS clusters

Alessandro Pacini[1], Davide Scano[1], Luca Valcarenghi[1], Andrea Sgambelluri[1], Alessio Giorgetti[2]
[1]Scuola Superiore Sant'Anna, Pisa, Italy - alessandro.pacini@santannapisa.it
[2]National Research Council of Italy (IEIIT-CNR), Pisa, Italy

2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) | 978-1-6654-7334-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/NFV-SDN56302.2022.9974775

***Abstract*—This demo presents an implementation of a resilient hierarchical synchronization system for SDN ONOS controller clusters. Two applications are designed and developed to be used respectively at child clusters and parent cluster to propagate network topology events over dedicated gRPC channels. Therefore, topological views of all the considered child domains are kept synchronized at the parent despite possible control plane failures. This work represents a novel solution specifically designed for hierarchical synchronization in ONOS clusters.**

***Index Terms*—Hierarchical, SDN, ONOS, Clusters, gRPC**

I. Overview

ONOS is one of the most widely used open-source Software Defined Networking (SDN) controllers [1]. It achieves high performance and scalability, while supporting the management of a large number of devices through different protocols (e.g., OpenFlow, P4, Netconf, etc.). Thanks to its modular architecture, applications can be easily developed and installed on top of it, thus extending the controller functionalities. ONOS also supports cluster configuration, where multiple instances act together to manage the same network. In this way, resiliency and scalability of the entire architecture are drastically increased. However, in this multi-instance configuration, cluster's instances have strict synchronization requirements to keep aligned the topology view. Thus, a rapid fail-over in case of data plane and control plane failures is achievable. For this reason, when it comes to handle a wide area network, a typical approach is to split it into multiple SDN domains. Each domain is managed by a single instance controller or cluster, where no one has the full topology view. In this context, two architectural designs can be found in the literature to share topology views among different SDN domains [2]. The first one is the flat design, in which each controller shares its local network view with one or more SDN controllers, typically using pub-sub frameworks [3]. In this way, each controller owns the global topology view while managing just its own local domain. The second one is the hierarchical design, where there are two types of instances: child and parent. Each child instance controls its domain, sharing its state with the parent one. The latter is the one which is able to reconstruct the entire topology view by combining the information coming from the children. In such scenarios, point-to-point communications are typically used among controllers [4].

This work has received funding from the European Union's Horizon 2020 research and innovation programme, B5G-OPEN Project, grant agreement No. 101016663. This work also received funding from the ECSEL JU project BRAINE (grant agreement No 876967). The JU receives support from the EU Horizon 2020 research and innovation programme and the Italian Ministry of University and Research (MUR).

In this work, an event-based synchronization mechanism for a hierarchical design is proposed. The system exploits two specifically developed applications to propagate network topology updates from children to parent over dedicated gRPC channels. Finally, the applications have been intended to exploit the ONOS distributed configuration, thus enabling resilient hierarchical synchronization among ONOS clusters. To the authors knowledge, this is a novel solution for specifically enabling ONOS clusters to a hierarchical design.

II. System Architecture and Workflow

The system relies on two developed applications:

- **Hierarchical Sync Child**, running on the ONOS child clusters;
- **Hierarchical Sync Parent**, running on the ONOS parent cluster.

Fig. 1 illustrates an example of system deployment, where two ONOS child clusters synchronize their topology views towards the ONOS parent cluster. For the Hierarchical Sync Child, the application is composed of two main parts: the *Listener* and the *Sender*. The *Listener* implements a mechanism for which events related to the managed domain are processed, parsed and stored in a distributed and reliable way. On the other hand, the *Sender* consumes those events from the storage while sending them over a gRPC channel towards the parent cluster, acting as client. Meanwhile, the Parent application includes two different parts: the *Receiver* and the *Publisher*. The *Receiver* takes the child events, received by the local running gRPC server, and saves them into a parent-dedicated distributed storage. The *Publisher* instead retrieves, reconverts, and publishes the events onto its local topology view (i.e., making the topology exploitable by ONOS core services) while preserving their generation order per child. Both applications use the synchronization mechanisms provided by ONOS to develop distributed applications. Indeed, any functionality implemented by each of them is actually replicated on each ONOS instance in the cluster. In this way, the system is able to seamlessly manage ONOS cluster instance failures, both at parent and children, without losing events. Further technical details about system's synchronization resiliency will be discussed during the demo. As for the gRPC channels, they have been developed to use unary and synchronous calls, exchanging properly converted ONOS event objects.

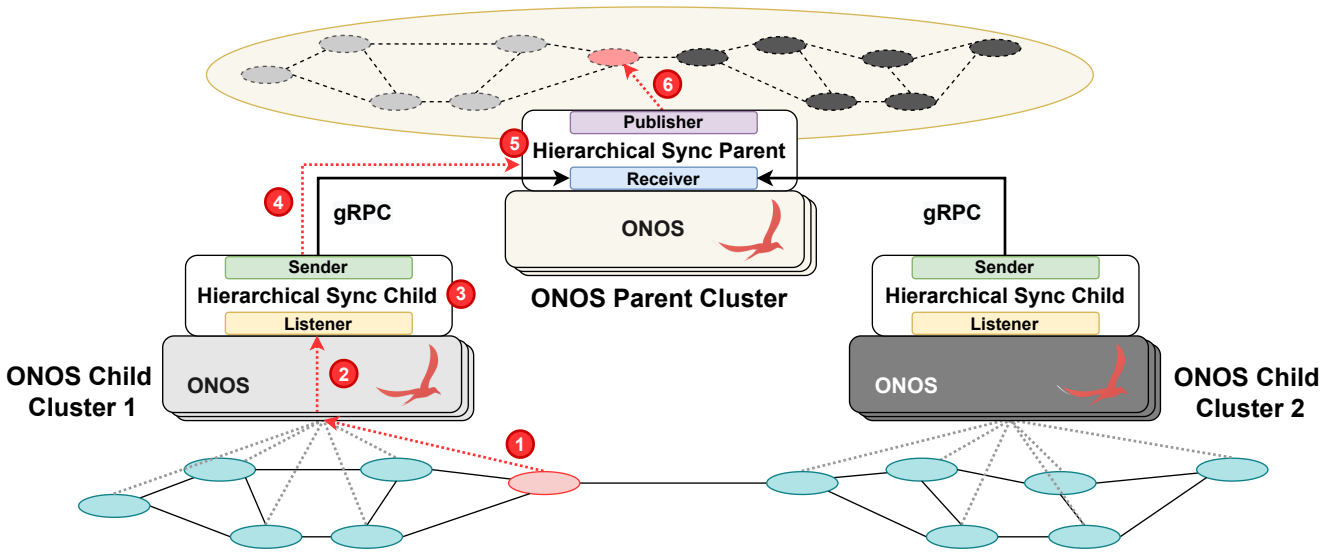

Fig. 1. High-level architecture view and synchronization workflow.

Fig. 1 also provides a visual representation of the entire workflow to be demonstrated (the numbers in the figure and here below indicate the sequence of events):

1) A network topology event (e.g., port down) occurs in the switch represented in red and controlled by ONOS child cluster 1;
2) The ONOS core dispatches the event within the cluster instances, ensuring a consistent view among them;
3) The Hierarchical Sync Child app:
   - Captures the event through its *Listener* component;
   - Sends it over the gRPC client within the *Sender* component.
4) The event propagates from the child to the parent cluster;
5) The Hierarchical Sync Parent app:
   - Receives the event at the gRPC server embedded in the *Receiver* component;
   - Reproduces the event into the parent view using the *Publisher* one.
6) All the parent cluster instances align with each other, and so with the latest topology view of the child.

III. Demonstration Walkthrough

This demonstration aims to show the potential of the proposed hierarchical synchronization system. Three ONOS clusters are used: two children and one parent. After installing the aforementioned applications on each cluster, a Mininet-based topology is used to emulate an OpenFlow network. The topology is divided into two SDN domains, each one managed by a different child. During this phase, the ONOS GUI is displayed for each cluster (e.g., children and parent). To this extent, it is possible to graphically show how fast the topologies of the child clusters synchronize with the parent. Indeed, as far as the elements are discovered by each child, they appear both at the child and at the parent view. The latter, in this case, combines together the views of the two domains.

Later, other events are manually generated by switching down network elements in the Mininet CLI on both SDN domains, further proving the responsiveness of the system that will update the parent view. Finally, within the same scenario, some ONOS instances (both at one of the children and at the parent clusters) will be deactivated emulating a failure in the control plane. This will demonstrate the resilience of the proposed implementation, which will continue to process events despite instance failures.

IV. Impact

In this demo, the implementation of an event-based hierarchical synchronization system for ONOS clusters is presented. The key aspect of this work is the full compatibility and integration with the ONOS architecture. Thanks to the transparency of the developed applications, it is possible to natively use the information of the child topologies at the parent side. This lays the foundation for a new set of parent applications that could be realized. Indeed, by exploiting the ONOS core services, hierarchical applications can be programmed to react to child events. Thus, the global view could be further exploited to apply both reactive and proactive cross-domain decisions.

References

[1] "ONOS." [Online]. Available: https://opennetworking.org/onos/
[2] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: A survey," *IEEE Access*, vol. 6, pp. 15 980–15 996, 2018.
[3] M. Gerola, F. Lucrezia, M. Santuari, E. Salvadori, P. L. Ventre, S. Salsano, and M. Campanella, "ICONA: A peer-to-peer approach for software defined wide area networks using ONOS," in *2016 Fifth European Workshop on Software-Defined Networks (EWSDN)*, 2016, pp. 37–42.
[4] D. Scano, A. Giorgetti, A. Sgambelluri, E. Riccardi, R. Morro, F. Paolucci, P. Castoldi, and F. Cugini, "Hierarchical control of sonic-based packet-optical nodes encompassing coherent pluggable modules," in *2021 European Conference on Optical Communication (ECOC)*, 2021.