Demonstration of High-availability communication based on Software-defined Networking

Thomas Pfeiffenberger and Ferdinand von Tüllenburg
Salzburg Research Forschungsgesellschaft mbH
Advanced Networking Center
email: thomas.pfeiffenberger@salzburgresearch.at
email: ferdinand.tuellenburg@salzburgresearch.at

***Abstract*—This demonstration addresses the subject of high-availability communication by utilizing software-defined networking (SDN) and Fog/Edge-based network function virtualisation (NFV).**

A particular focus is given on high available communication conceptually inspired by the parallel redundancy protocol (PRP) and high-availability seamless redundancy protocol (HSR). The usage of SDN and Fog/Edge-based NFV allows a more flexible management of high available communication infrastructure. It corresponds to the specific needs of certain applications and providing the necessary deep packet inspection capabilities at the right spot with sufficient performance.

The demonstration shows several test cases in context of a high-availability communication enabled SDN in an multi-application environment.

I. Introduction

High-availability communication solutions tend to protect ICT-systems against failures of the underlying network components. The requirements for high-availability communication are prevalent in critical infrastructures such as machine-to-machine communication in power system control or manufacturing systems. High-availability communication is usually related to reduce packet drops to a minimum that a certain application is able to operate without malfunction. Different application may require different levels of high-available communication.

In order to guarantee high available communication technologies such as the parallel redundancy protocol (PRP) or the high-availability seamless redundancy protocol (HSR) [1] has been developed. These technologies make use of redundant but strictly separated communication networks by sending duplicated data packets simultaneously via the redundant networks. In order to make the packet duplication transparent to applications, duplicates are filtered out before final delivery. Usually implemented as dedicated communication networks with frequent use of proprietary hardware making these solutions somewhat static and tightened to specific applications such as manufacturing control.

The ongoing digitisation of manufacturing systems - not least because of the developments of the industrial Internet of Things (IoT) and Industry 4.0 - however, increasingly requires converged communication networks. Such networks consist of a single infrastructure, which can be flexibly adapted to various communication requirements of different applications ranging from loss-sensitive manufacturing control to time critical real-time analytics and less demanding business intelligence applications. A main building block of converged networks is network programmability as proposed by the software-defined network (SDN) and network function virtualisation (NFV) approach.

This demonstration shows how SDN and NFV can be utilized to implement a high-available communication solution. While adapting the general concepts of solutions such as HSR and PRP, the proposed solution has three properties especially fitting to the vision of future converged communication networks:

- Completely based on standardized SDN / OpenFlow technology.
- Support for arbitrary network topologies and dynamic network reconfiguration.
- Support for application-specific traffic treatment.

The SDN-based high-availability communication solutions demonstrated in this work operates in a proactive manner aiming at a zero-loss policy and should be seen in contrast to solutions which minimize communication failures in a reactive operation. Several solutions in this direction has been described in our previous work [2].

II. High Availability Communication with SDN

As described earlier the general concept of the demonstrated solution is borrowed from HSR and PRP. Data packets are duplicated and sent via disjunct paths towards its destination in order to avoid packet loss due to failures of network devices. At a certain point (fork point) within the network packets are duplicated and forwarded using multiple disjunct paths. The number of equipment failures which can be intercepted depends on the number of used disjunct links used for parallel sending. At a further point in the network, where disjunct paths are joining (junction point) duplicates of already arrived packets are dropped in order to guarantee duplication transparency for applications. Duplicates are usually identified using distinguishing marks (such as serial numbers) and inter-arrival time intervals: Only the first packet with a certain distinguishing mark arriving within a time interval is forwarded.

In this demonstration the programmability of a SDN/OpenFlow network is utilized to implement the high-availability communication concepts on an arbitrary network topology. In contrast to HSR and PRP, where the redundant networks are physically separated (without any interconnections between them), SDN based networks allows a separation into logical network slices with similar properties as physical separated networks. In this way disjunct communication paths between certain end-devices can be nearly instantaneous created and dynamically reorganised.

In view of future developments in direction of converged networks this can be seen as a certain improvement, as already has been shown for time-triggered manufacturing networks [3]. Furthermore, packet selection capabilities of OpenFlow based on header field contents allow for duplication treatment specific to certain applications and use cases. While in pure HRS/PRP all traffic types traversing the network are handled in the same way (in particular sent on redundant paths in parallel), OpenFlow allows for handling different traffic types in distinct ways.

Traffic flows of certain applications can be provided with high availability e. g. for machine-to-machine communication in a manufacturing environment, while other traffic types, for instance for business intelligence applications can be treated in a best-effort manner. Finally, both properties mentioned above allow for freely arranging fork points and junction points within the network.

In case of HSR / PRP duplication removal is done by receiving nodes or a dedicated redundancy box (RedBox). In an SDN, however, this functionality is a candidate to be implemented as virtualised network function. In doing so, the filtering functionality is not bound to certain network nodes and can be used at arbitrary points within a SDN domain. However, duplication detection uses deep packet inspection for analysing distinguishing marks of individual packets requiring high computing power.

While sufficient computing capacities could certainly be provided by cloud based NFV environments, the inevitably appearing latency for sending each incoming packet to the cloud is a not negligible obstacle. A solution to this problem can be found in fog or edge computing approach, where the required deep packet inspection functionality is shifted to a computing environment located directly at network nodes like a switch having sufficient computing capabilities.

While HSR and PRP uses serial numbers as distinguishing mark for duplication detection (added to the padding area in Ethernet frames), the implemented proof-of-concept algorithm works without frame manipulation by instead evaluating the entire payload of each received packet. The payload of an incoming packet is compared with the payloads of formerly received packets stored in memory - in a first step by size and second by content.

If a received packet payload matches with a stored packet payload, the newly arrived packet is considered as duplicate and silently dropped. Otherwise, the packet's payload is stored in memory and then forwarded towards it's designated receiver. Packet payloads are stored in memory for a configurable time after which the payload will be removed from memory.

This goes in line with implementations of HSR and PRP, taking account for the assumption that duplicate packets arrive within a relatively short time interval at the junction point. Thus, the amount of stored packets can be significantly reduced in order to improve lookup performance and make applications unconcerned of the duplication process, which intentionally send equal packets in (fixed) time intervals (such as heartbeats).

III. Demo Setup

In this paper, a demonstrator for a high reliable communication solution in a sliced SDN network is introduced. The goal of this demonstration is to show how the combination of sliced SDN, NFV and FOG/Edge computing can be a potential solution to provide a high reliable communication in one shared communication infrastructure.

As shown in Fig. 1 the demo setup consists of a ONOS based SDN network, a PC based IP traffic generator [PC1] used as an sensor, a traffic receiving unit [PC2] with an Adruino providing some traffic quality indicator and monitors to visual packet behaviour in the SDN network. The Fork point device at the ingress point of the network duplicates all packets from a defined traffic class and sends the duplicated packets on different paths to the SDN network. At the egress point, the Junction point device filters and drops duplicated packets.

IV. Demonstration

The demonstration shows a high-availability communication in a common SDN-based network. Network applications with two different Quality of Service (QoS) requirements share one network.

One application is considered as critical and is highly sensitive to packet loss and reordering, while the second application is less critical and resistant against packet losses or (short time) link losses.

For both applications different high-availability methods are shown in the demonstration. While the critical application uses the duplication based approach described in this paper, for the less critical application an intent based high-availability method is applied which operates in a reactive way onto network failures. As soon as a network failure hindering the application to communicate has been detected by the network monitoring system, a failover process computes and automatically installs a new path through the network. In the demo, network failures are induced by detaching network cables and power off or shutdown network devices.

For application specific high-availability management a tool is presented, which allows to management, control and monitoring both network intents and duplication based redundancy. Furthermore, the demonstrated tool shows the operation of the duplication detection network function in an edge computing approach running directly at the switching device providing

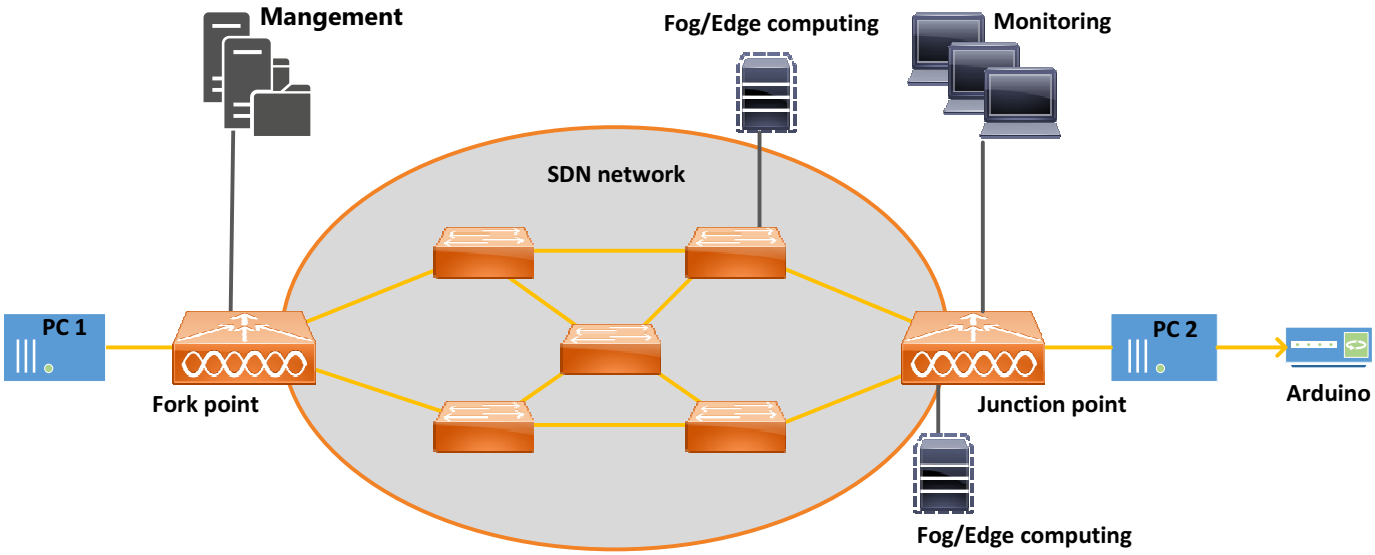

Fig. 1: Architecture

the junction point. The tool is integrated in the ONOS network controller [4].

For both, intent-based and duplication-based high-availability the effects on the respective applications are shown with reference to occurrence of packet loss, reordering or duplication and even timing-constraints. Multiple test cases with varying packet sizes and sending intervals will be used for demonstration.

Live-evaluation of the demonstration is provided by network monitoring tools (e. g. bandwidth utilization per link and flow) and an Arduino experiment representing an actor of a critical cyber-physical-system sensitive to duplicated, lost or reordered packets as well as for delayed data arrival.

The Arduino experiment consists of an Arduino Uno and a breadbord with three coloured LEDs used as visual indicator of the communication quality. The green LED visualizes correctly ordered and in-time data reception, while the blue LED indicates out-of-time (but otherwise well) data reception and the red LED indicates packet loss, reordering or duplication.

A control application running at the data receiver processes the data packets coming from the network and decides if the packet has been received without loss, reordering or duplication. This decision is taken by evaluating the arrival order of packets based on packet numbers transported in the IP payload of the packets. Depending on the occurrence of gaps or interchanges in the arrival order either a control signal indicating valid or invalid reception is transmitted to Arduino. In the first case Arduino will light the green LED in the second case the red LED will be lighted for a period of 200 ms.

Compliance to the timing constraint, in contrast, is evaluated by the Arduino board directly. For the showcase it is assumed that Arduino expects a continous stream of control signals with a maximum delay between two successive signals (e. g. 1.2 ms at a sending interval of 1 ms). In case this condition is violated the Arduino will light the blue LED for a period of 200 ms.

Acknowledgment

The work presented in this paper is part of the OPOSSUM project funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT).

References

[1] International Electrotechnical Commission, "IEC 62439-3:2016 - Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)," Mar. 2016.

[2] F. von Tüllenburg and T. Pfeiffenberger, "Layer-2 Failure Recovery Methods in Critical Communication Networks," in *ICNS 2016: The Twelfth International Conference on Networking and Services*. Lisbon: IEEE, Jun. 2016, pp. 1–4.

[3] M. Herlich, J. L. Du, F. Schorghofer, and P. Dorfinger, "Proof-of-concept for a software-defined real-time ethernet," in *21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2016, Berlin, Germany, September 6-9, 2016*, 2016, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ETFA.2016.7733605

[4] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "Onos: Towards an open, distributed sdn os," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2620728.2620744