Demo: Scalable and Reliable Software-Defined Multicast with BIER and P4

Wolfgang Braun, Joshua Hartmann, Michael Menth
University of Tübingen, Department of Computer Science, Germany

***Abstract*—We present an architecture for software-defined multicast that provides scalable and reliable multicast services. We consider the Bit Indexed Explicit Replication (BIER) architecture and Traffic Engineering for BIER (BIER-TE) for packet transport to mitigate scalability concerns of traditional IP multicast.**

We briefly discuss the advantages of an SDN-based multicast architecture leveraging BIER and discuss implementation options: OpenFlow and the P4 language. We present our prototype based on P4. We implement traditional IP multicast and several BIER variants that may be deployed simultaneously with the prototype. We present various demo scenarios that show the feasibility and the advantages of this architecture.

***Index Terms*—Bit Indexed Explicit Replication, Multicast, Resilience, Scalability**

I. Introduction and Scientific Rationale

The Internet Engineering Task Force (IETF) currently discusses BIER [1] to solve issues with traditional IP multicast deployments. These issues include various scalability concerns for applications like IPTV and L3VPNs. Moreover, frequent dynamic multicast tree updates cause operational issues for over-the-top services [2], [3]. With BIER, ingress switches encapsulate packets with a novel BIER header that encodes the egress nodes. Transit nodes forward BIER packets only using the BIER header. The forwarding entries required for BIER are topology-dependent and are the same for all multicast flows in the network. Therefore, many scalability concerns can be resolved using BIER. In addition, multicast tree updates can easily be frequently supported because only ingress switches have to be updated when multicast subscriptions change. BIER-TE [4] allows to explicitly define the path of the multicast tree by encoding the links and egress nodes in the header. BIER-TE also provides Fast Reroute (FRR) [5] to provide reliable multicast services.

Currently, implementations for BIER are standardized for MPLS [6], OSPF [7] and IS-IS [8] to integrate BIER in current IP/MPLS networks. However, this most likely will require hardware support for BIER and cause additional CAPEX through the need of new BIER-capable routers. Software Defined Networking (SDN) and OpenFlow [9] provide means to easily implement new features and services by separating data and control plane. Since version 1.2, OpenFlow supports different protocol headers, e.g. BIER, using the OpenFlow Extensible Match (OXM). Multicast and FRR require group tables in OpenFlow. However, different group table types are required and cannot be mixed. In addition, we think that the pipeline prior OpenFlow 1.5 is too restricted to support BIER and BIER-TE. OpenFlow 1.5 is not implemented in current OpenFlow switches and BIER will most likely require these optional features that may not be supported by all OpenFlow switches in the future.

The authors acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG) under grant ME2727/1-1. The authors alone are responsible for the content of the paper.

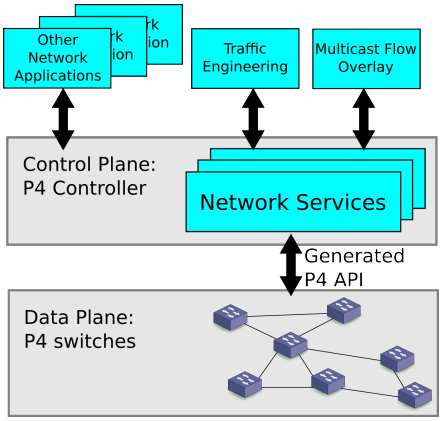

Fig. 1: Software-defined multicast architecture with P4.

Therefore, we consider P4 [10] as southbound interface for an SDN-BIER approach. P4 allows both the programmability of the data plane with regard packet format and to packet control through the switch's pipeline. New packet headers can be defined and P4 programs allow for complex pipeline operations. The intent of this work is to show that the P4 language is able to implement a scalable and reliable SDN-based multicast architecture that simultaneously supports traditional IP multicast, BIER, BIER-TE, and BIER-TE FRR.

II. Architecture and Testbed

In this section we discuss a generic SDN-based multicast architecture. We focus on the components and their functionalities. Then, we discuss our prototype implementation and the testbed.

A. Architecture

The generic architecture is illustrated in Figure 1. It is based on the typical SDN structure. The data plane consists of P4 switches that are controlled by one or more controllers. The controllers configure the switches using the API that is generated by the P4 compiler of the P4 source code. The

controllers interact with the so called Multicast Flow Overlay (MFO) that manages multicast subscriptions and can be provided by approaches such as defined in the RFCs 6513 or 6514. In addition, we propose the use of a Traffic Engineering (TE) component that provides optimized paths for BIER-TE. FRR can be installed if traffic should be protected. Note that BIER multicast technologies differ with regard to complexity, state overhead, and required resources [11]. The controller will install either traditional IP multicast, BIER or BIER-TE depending on the specific use case and its requirements given by the MFO, TE, or other network applications.

B. Prototype Implementation

We provide a P4 implementation of BIER and BIER-TE. We release P4 source code of our multicast router as open source under the Apache License. The software is available online [12]. We use the same BIER header proposed for BIER MPLS encapsulation [6] for our implementation but embed BIER and BIER-TE into an Ethernet infrastructure instead of MPLS. We use the P4 reference software switch called "behavioral model" (bmv2) [13]. The reference switch has similar importance to P4 as OpenVSwitch to OpenFlow and is - as far as we know - the only P4 switch available at this time. Our testbed runs in a virtual machine with Ubuntu 14.04. We emulate various network scenarios in Mininet 2.3. The demo scenarios of our prototype are validated using multicast test tools developed by University of Virginia's Multimedia Networks Group [14].

We use a static network topology in our demonstration which is illustrated and controlled by a GUI written in Python and Qt. The GUI installs forwarding entries in the P4 switches using the Command-Line Interface (CLI) which is automatically generated of the P4 source code. The GUI acts as SDN controller but lacks some features such as topology detection, etc.

III. Demo Scenarios

In this section we describe the demo scenarios. All use cases can be simultaneously executed in the demonstration.

A. Traditional IP Multicast

We implement traditional IP multicast in the P4 switches. Each switch contains a mapping of multicast addresses to corresponding outgoing ports. Every switch that is part of the multicast tree requires a specific IP multicast table entry. Packets are duplicated correctly at branching points.

B. BIER Multicast

We configure the Bit Indexed Forwarding Table (BIFT) in all P4 switches according to the BIER IETF draft [1]. For each multicast tree, one entry is installed at the ingress switch. The entry contains an action that encapsulates the BIER header which contains the egress nodes of the multicast tree. The multicast packet is forwarded along the shortest path tree towards the destinations using the BIFT entries. The BIER header is removed at the egress switches and the original packet is forwarded to the receiving host.

C. Explicit Paths Using BIER-TE

We configure the BIER-TE tables similarly to BIER multicast according to the BIER-TE draft. The ingress nodes encapsulate BIER headers that contains the links to traverse and the egress nodes. Packets do not have to follow the shortest paths. We leverage the EtherType field to distinguish IP, BIER, and BIER-TE multicast traffic to match packets against the corresponding forwarding tables in the P4 program.

D. Reliable Multicast Using BIER-TE FRR

BIER-TE FRR is based on rewriting the forwarding information contained in the BIER header in such a way that the packets are locally rerouted to the next-hops in case of failures. There are three different mechanisms proposed to implement the backup paths. We implemented BIER-in-BIER encapsulation (BBE) in our prototype for local reroute. We omit the two others variants because they are generally outperformed by BBE [11].

We configure the BIER-TE Adjacency Forwarding Table (BTAFT) to protect links in the network. We show that different multicast trees are rerouted correctly using the same BTAFT entries when a link failure occurs.

References

[1] I. Wijnands, E. C. Rosen, S. Aldrin, T. Przygienda, and A. Dolganow, "Multicast using Bit Index Explicit Replication," Internet-Draft, Jul. 2016.

[2] G. Shepherd, A. Dolganow, and A. Gulko, "Bit Indexed Explicit Replication (BIER) Problem Statement," Internet-Draft, Apr. 2016.

[3] C. Bestler, N. Kumar, R. Asati, M. Chen, X. Xu, A. Dolganow, T. Przygienda, A. Gulko, D. Robinson, and V. Arya, "BIER Use Cases," Internet-Draft, Jul. 2016.

[4] T. Eckert, G. Cauchie, W. Braun, and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE," Internet-Draft, Jul. 2016.

[5] ——, "Fast ReRoute (FRR) Extensions for BIER-TE," Internet-Draft, Jul. 2016.

[6] I. Wijnands, E. Rosen, A. Dolganow, J. Tantsura, and S. Aldrin, "Encapsulation for Bit Index Explicit Replication in MPLS Networks," Feb. 2015.

[7] P. Psenak, N. Kumar, I. Wijnands, A. Dolganow, T. Przygienda, J. Zhang, and S. Aldrin, "OSPF Extensions For BIER," https://datatracker.ietf.org/doc/draft-ietf-bier-ospf-bier-extensions/, Oct. 2015.

[8] L. Ginsberg, A. Przygienda, S. Aldrin, and J. Zhang, "BIER support via ISIS," https://datatracker.ietf.org/doc/draft-ietf-bier-isis-extensions/, Oct. 2015.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communications Review*, vol. 38, no. 2, pp. 69–74, 2008.

[10] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming Protocol-Independent Packet Processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656890

[11] W. Braun, M. Albert, T. Eckert, and M. Menth, "Performance Comparison of Resilience Mechanisms for Stateless Multicast Using BIER," in *IFIP/IEEE Symposium on Integrated Management (IM)*, May 2017.

[12] W. Braun and J. Hartman, "P4 Bit Forwarding Router (p4-bfr)," https://bitbucket.org/wb-ut/p4-bfr, 2017.

[13] "Behavioral model repository," https://github.com/p4lang/behavioral-model, Jan 2017. [Online]. Available: https://github.com/p4lang/behavioral-model

[14] "Tools for multicast testing (msend and mreceive)," https://github.com/troglobit/mtools, Jan 2017. [Online]. Available: https://github.com/troglobit/mtools