BNG-HAL: A Unified API for Disaggregated BNGs

Rubens Figueiredo
BISDN GmbH
PhD Student
rubens.figueiredo@bisdn.de

Andreas Kassler
Karlstad University
Supervisor
andreas.kassler@kau.se

***Abstract*—Network Functions have traditionally been deployed as software packages inside virtual machines or containers. In order to make them cloud-native and exploit the low-cost, scalability and resiliency of the cloud, Network Functions need to be disaggregated. Such disaggregation enables flexible deployment and resource allocation policies for critical components of a Network Function. However, growing interest of vendor neutrality and interoperability from vendors and operators, requires standardisation of well-defined interfaces. In this paper, we propose an abstraction layer for the disaggregation of a broadband network gateway (BNG). In the proposed architecture, this layer abstracts the BNG functions, and provides a stable translation target for configuring and managing subscriber sessions by adding forwarding rules or specifying Quality of Service parameters.**

1. Introduction

A Broadband Network Gateway (BNG) is an essential access network technology for providing broadband connectivity. The BNG provides per-subscriber network access and services, and is the point in the network splitting the access network to the routed Layer 3 network. The Broadband Forum (BBF) specifications TR-101 [1] and TR-178 [2] define the requirements for a BNG, with the high level requirements including VLAN support/ differentiation, Quality of Service (QoS) enforcement and Traffic Management.

Cloud-native engineering concepts influence the networking services currently being rolled out. This design pattern advocates the usage of cloud computing to achieve flexibility to cope with new requirements, scalability to adjust to dynamic changes in workload and resiliency to not fail under presence of failures. The cloud-native concept is best supported by disaggregated performance, which aims to break down the whole functionality into individual micro-services that can be flexibly deployed. It supports fine-grained resource allocation to eliminate potential design bottlenecks and flexible hardware acceleration of time-critical tasks.

To achieve disaggregation, Software Defined Networking (SDN) advocates for control and user planes separation, allowing the independent provisioning of both elements and improved programmability [3]. The SDN control plane has a centralised view of the entire managed network, and such knowledge can be leveraged to reconfigure the network or optimise traffic steering policies [4]. The disaggregated BNG (DBNG) supports running the DBNG control plane (DBNG-CP) functions in a central control block, while controlling multiple user plane (DBNG-UP) instances, deployed in hardware or software, which allows for better scaling and redundancy of each plane [5]. The different platforms involved in the deployment of a DBNG-UP introduces a challenge where the features are equivalent, but the forwarding pipelines may differ, which impacts the capability to handle subscriber growth and their increasing bandwidth demands.

In this work, we explore the challenges on the design and deployment of disaggregated BNGs. The BNG-HAL, a DBNG user plane abstraction layer and our contribution for the DBNG architecture, is a component integrated between the CP and UPs, abstracting the common features between UPs. It abstracts functionality by including a target-independent Application Programming Interface (API) that programs and manages DBNG UPs. We analyse and propose solutions for tackling challenges related to the abstraction of the DBNG-UP that range from splitting the forwarding and management control channels to modelling and supporting complex traffic management features.

This paper is structured as follows. In section 2, we present a literature review on the available standards and approaches for DBNG architecture. In section 3, we formulate the problem and state the main areas of contribution of the work. In section 4, we present the main challenges encountered while adapting the API for abstracting the DBNG-UP functionality. In section 5, we provide an overview of the work so far developed, summarise the contributions of the work and the future steps.

2. Background

The BBF-TR 459 [6] specifies the architecture and requirements for a DBNG, defining both the network functions in the CP and a set of UPs, and the interfaces between them. Figure 1 shows a simplified BBF TR-459 DBNG, with the split of the DBNG-CP and DBNG-UPs functions and a generic representation of the existing interfaces between these two layers. These interfaces are

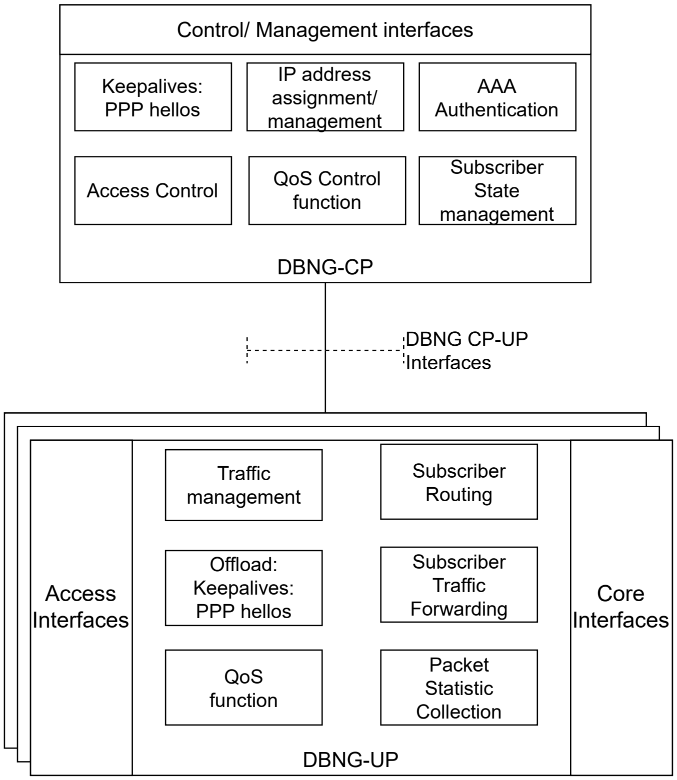

Figure 1: Simplified BBF TR 459 core CUPS separation, and respective functions [6]

- **Control Packet Redirection Interface (CPRi)**: tunnels control packets to and from the UP and CP;
- **State Control Interface (SCi)**: used by the CP to configure the UP to route/forward packets. The proposed protocol for this interface is the Packet Forwarding Control Protocol (PFCP) [7];
- **Management Interface (Mi)**: the CP and UP push and retrieve platform management configuration, which include QoS and ACL templates. The definition of this interface is out-of-scope for the BBF TR-459.

The DBNG supports multiple UPs being controlled by the centralised CP. For supporting interoperability between multiple dataplanes, the management and operation interfaces [8] must be defined. To support this effort, OpenConfig[1] and BBF[2] define vendor-neutral datamodels to model the available resources. Specifically on access networks, [9] [10] define YANG models [11] to manage interfaces, layer 2 forwarding, or statistics retrieval. NETCONF manages, retrieves and uploads configuration data to and from network devices, and combined with the YANG data modelling language allows for a domain specific configuration management protocol [12].

1. https://openconfig.net/
2. https://www.broadband-forum.org/

The DBNG architecture expects that the traffic is shaped in order to ensure bandwidth based on customer contracts [13]. The Quality of Service (QoS) requirements separate the traffic into different classes, and each BNG solution is expected to have multiple ways of representing these, with different classes of service being necessary for representing service level agreements and subscriber bandwidth allocations. Therefore, models representing these traffic control profiles must be highly flexible to allow each operator to manage their system. Depending on the ISP network structure, multiple traffic scheduling layers can be present with hierarchical relationships among each other. This complex traffic shaping is named Hierarchical Quality of Service (HQoS). To model this relationship between levels of service, standardised data models [9] can be applied.

In Kundel et al. [13], and accompanying implementation, the Unified Control Plane [14] is presented as a generic API for a DBNG. Defined are generic actions that a BNG operator has to do to provision customers.

3. Problem Statement

Supporting multiple UPs possibly implemented on different targets requires that the PFCP-UP agent be made aware of the platform-specific semantics for routing/ forwarding/ managing configuration. This approach makes the PFCP agent complex when considering integration to the platform target, thus emphasising the need for abstracting commonality between the DBNG-UPs.

The complexity of DBNG management limits the adoption of vendor neutral solutions, due to the tie of management systems to the target network and their typical proprietary nature [15]. As such, management of network resources requires their abstraction, where the necessary set of information is synthesised and serialized over standardised datamodels. Traffic shaping and scheduling is as well a central feature of the DBNG, and the diverse dataplane architectures have different APIs to configure traffic management along with different mechanisms to enforce it, which may result in heterogeneous deployments.

Hypothesis: Abstracting the functions of the DBNG-UP and synthesising them in a HAL between the CP and UP defines a clear target to program the dataplanes. This added component complements the DBNG to operate in a vendor neutral way, supporting multiple target pipelines and more flexible setups.

Research Questions: In this work we aim to explore the following questions. (Q1) How is the DBNG able to cope with multiple target dataplanes? (Q2) What are the limitations for the existing architectures and interfaces between CP and UPs? (Q3) How to minimise the integration effort of new platforms/ functions to the DBNG? (Q4) How adaptable is the proposed HAL to newer northbound protocols?

4. Challenges

While PFCP abstracts the actions for programming UP traffic forwarding, the ambiguous pipeline model and hard integration workflows impact interoperability [16]. Furthermore, PFCP is not able to model the hierarchical relationships and different levels of service of the HQoS parameters. With PFCP Quality Enforcement Rules (QERs), pre-provisioned rules in the UP can be referred to. This pre-provisioning of the HQoS templates is a management action. The QER then allows for controlling the flow of traffic belonging to a certain service or enforcing the maximum QoS defined at multiple levels, among other features. For example, QER can describe the maximum (MBR) or guaranteed (GBR) bitrate for upstream and/ or downstream [7] as well as a QoS flow identifier.

In order to abstract the actions from the UP, the Unified Control Plane [14] is designed as a generic API for a DBNG-UP. While the functions implemented in [14] are enough to fulfil their requirements, this API must be extended to support all of the actions of a DBNG. Notably, actions to set the control packet redirection rules and Quality of Service (QoS) parameters are missing. Additionally, the parameters to pass to each function call must also be analysed by extracting the necessary information elements (IE) from PFCP and mapping these to the abstract actions.

While the BBF TR-459 extensively defines the workflow and necessary protocol extensions to manage BNG subscribers over the SCi, the management interface is not properly defined. For the implementation of this interface, protocols such as NETCONF/ RESTCONF or gNMI can be used. The models present in [17] support specific requirements on access network equipment. By configuring the different HQoS parameters with models such as `bbf-qos-policies` or `bbf-qos-enhanced-scheduling` [17], and correlating them with the per-session PFCP rules, extreme fine-grained control over subscriber traffic could be achieved.

5. Current Status

The challenge of adapting an abstract forwarding protocol to multiple targets that might not support a given protocol can be solved by a Hardware Abstraction Layer (HAL) [18]. With the HAL, a single Control Plane can control multiple targets, even if they might not be supporting a certain protocol. We propose the BNG Hardware Abstraction Layer (BNG-HAL), which reduces the complexity of managing a BNG by presenting a set of features that removes the dependency on specific protocols.

The BNG-HAL architecture is depicted in Figure 2, where the SCi and Mi are terminated in the BNG-HAL layer. All necessary operations are then assembled for management and state control, translating them to the desired hardware target. Terminating both these interfaces in the BNG-HAL ensures that all configuration will be synchronised between the CP and UP by maintaining state in this layer.

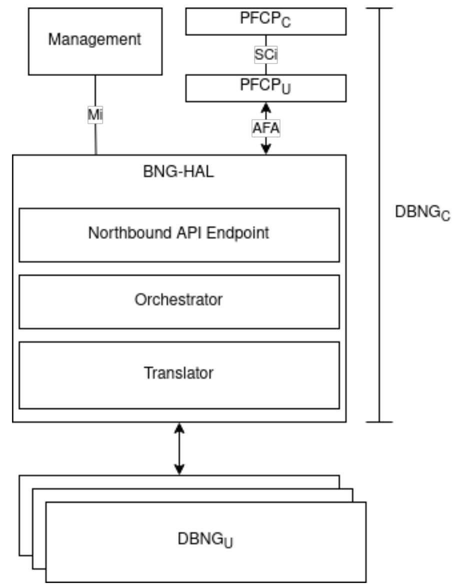

Figure 2: Logical components of the BNG-HAL

5.1. Proposed architecture

5.1.1. Abstract Forwarding API (AFA). By extending the Unified Control Plane [14] by methods that program the control packet redirection rules and setting of the HQoS parameters, the AFA is able to abstract the DBNG user plane actions. The AFA is designed to be abstract, accept the protocols defined in the BBF TR-459, and provide an easy interface for extending the API as necessary.

5.1.2. Translator. The translator module adapts the AFA calls to the appropriate hardware target. This module is composed of independent hardware drivers, each maintaining the translation semantics from the abstract API calls for their resources, and translating the configuration calls to platform-specific semantics. For example, when accelerating the DBNG-UP on a P4 programmable hardware device [19], the translator would map the AFA calls to P4Runtime [20] methods to install the proper match/action rules and trigger platform specific traffic manager configuration APIs.

5.1.3. Orchestrator. The DBNG is expected to support multiple UP instances, with each having the possibility of being formed by multiple independent modules. Orchestration plays a central role in the BNG-HAL by managing the hardware resources executing actions such as reporting failures, ensuring correct configuration and restoring the initial state of hardware components. The orchestrator stores state on the subscriber sessions and management information, being able to provide the necessary platform semantics for programming each UP.

5.2. Initial Evaluation

For evaluation of the proposed architecture we implement the BNG-HAL and perform functional and performance-related tests in our testbed setup. We defined the AFA with the abstracted actions to manage the customer sessions, install routing rules and setup the traffic shaping profiles.

We defined the intended hardware targets to provide translation for, by analysing the current existing offers of DBNG-UP, such as software solutions like DPDK or VPP, hardware solutions like the Tofino programmable switches or Broadcom, and finally mixed solutions, like Tofino and FPGA or smartNIC and DPDK. For initial implementation, the DPDK based implementation of the DBNG-UP was chosen as the target platform. The definition of the translation layer drivers then followed, by analysis of the existing control interfaces for this target.

For experimentation of the entire stack we must test the behaviour of the abstraction layer with multiple UP instances running simultaneously, and gather data for performance evaluation of the solution such as data plane specific metrics including packet forwarding latency and throughput as well as control plane related performance parameters that can identify the performance of the HAL.

5.3. Further steps

The next steps include the implementation and proof-of-concept of the proposed work, where multiple UPs with different architectures are used. One of the main issues to be explored is to analyse the behaviour of the BNG-HAL architecture in the case control-plane is under pressure due to e.g. micro-bursts caused by power outages and major reconnections.

6. Acknowledgments

We would like to thank Ricardo Santos and Mario Kind for providing valuable feedback and reviewing this work. We thank the support provided by Hagen Woesner, the team at BISDN as well as Hans-Jörg Kolbe and the team from Deutsche Telekom AG for providing feedback during development of this work.

References

[1] T. Anschutz, Cohen, Amit, and Shrum, Ed, "TR-101 Migration to Ethernet Based Broadband Aggregation," no. 2, p. 101, Jul. 2011. [Online]. Available: https://www.broadband-forum.org/download/TR-101_Issue-2.pdf

[2] "TR-178 Multi-service Broadband Network Architecture and Nodal Requirements," BBF, Tech. Rep. TR-178, Sep. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-178.pdf

[3] "TIP OOPT Open BNG Technical Requirement," Telecom Infra Project, Tech. Rep. [Online]. Available: https://assets2.brandfolder.io/bf-boulder-prod/jscc2vv6vkcn2hzhqm8xzcf/v/34839280/original/TIP%20OOPT%20Open%20BNG%20Technical%20Requirements%20v1.0.docx.pdf

[4] S. Sridharan, "A Literature Review of Network Function Virtualization (NFV) in 5G Networks," *International Journal of Computer Trends & Technology*, vol. 68, no. 10, pp. 49–55, Oct. 2020. [Online]. Available: http://ijcttjournal.org/archives/ijctt-v68i10p109

[5] O. Ts, "SDN Enabled Broadband Access," p. 68, 2021. [Online]. Available: https://opennetworking.org/wp-content/uploads/2021/03/ONF-Reference-Design-SEBAv2.0FINAL3.pdf

[6] Wan, Kenneth, "TR-459 Control and User Plane Separation for a disaggregated BNG," Jun. 2020. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-459.pdf

[7] P. Schmitt and K. Kymalainen, "LTE; 5G; Interface between the Control Plane and the User Plane nodes," ETSI/ 3GPP, Technical Specification TS 129 244, Nov. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129200_129299/129244/15.10.00_60/ts_129244v151000p.pdf

[8] N. Sambo, K. Christodoulopoulos, N. Argyris, P. Giardina, C. Delezoide, A. Sgambelluri, A. Kretsis, G. Kanakis, F. Fresi, G. Bernini, H. Avramopoulos, E. Varvarigos, and P. Castoldi, "Experimental Demonstration of a Fully Disaggregated and Automated White Box Comprised of Different Types of Transponders and Monitors," *Journal of Lightwave Technology*, vol. 37, no. 3, pp. 824–830, 2019.

[9] W. Yang, J. Chen, and T. Carey, "YANG Modules for Network Map & Equipment Inventory," no. 1, p. 16, Jun. 2021. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-454_Issue-1.pdf

[10] "Common YANG Modules for Access Networks." [Online]. Available: https://www.broadband-forum.org/technical/download/TR-383_Amendment-4.pdf

[11] M. Björklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," Oct. 2010, issue: 6020 Num Pages: 173 Series: Request for Comments Published: RFC 6020. [Online]. Available: https://rfc-editor.org/rfc/rfc6020.txt

[12] J. Schönwälder, M. Björklund, and P. Shafer, "Network configuration management using NETCONF and YANG," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 166–173, 2010.

[13] R. Kundel, L. Nobach, J. Blendin, W. Maas, A. Zimber, H. Kolbe, G. Schyguda, V. Gurevich, R. Hark, B. Koldehofe, and R. Steinmetz, "OpenBNG: Central office network functions on programmable data plane hardware," *International Journal of Network Management*, vol. 31, no. 1, Jan. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nem.2134

[14] L. Nobach, C. Cascone, R. Kundel, and L. Prete, "p4se." [Online]. Available: https://github.com/opencord/p4se

[15] N. Sambo, A. Giorgetti, A. Sgambelluri, P. Castoldi, and L. Valcarenghi, "Software-Defined Reconfigurability, White Boxes, and Abstraction." in *ONDM*, 2019, pp. 15–25.

[16] C. Cascone, M. Kind, and C. Stevens, "Tassen: Towards a Next-Generation BNG CUPS API," Jul. 2020. [Online]. Available: https://opennetworking.org/wp-content/uploads/2020/07/Cascone_Kind_Stevens-2020-Broadband-Spotlight-Slide-Deck.pdf

[17] N. Hancock, L. Pauwels, J. Boyd, S. Padmanabhan, and S. Ooghe, "TR-383 Common YANG Modules for Access Networks," BBF, TR 383, Jun. 2021. [Online]. Available: https://www.broadband-forum.org/technical/download/TR-383_Amendment-4.pdf

[18] D. Parniewicz, B. Belter, E. Jacob, K. Pentikousis, R. Doriguzzi Corin, L. Ogrodowczyk, M. Rashidi Fard, J. Matias, M. Gerola, V. Fuentes, U. Toseef, and A. Zaalouk, "Design and implementation of an OpenFlow hardware abstraction layer," in *Proceedings of the 2014 ACM SIGCOMM workshop on Distributed cloud computing - DCC '14*. Chicago, Illinois, USA: ACM Press, 2014, pp. 71–76. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2627566.2627577

[19] The P4 Language Consortium, "P416 Language Specification," May 2021. [Online]. Available: https://p4lang.github.io/p4-spec/docs/P4-16-v1.2.2.html

[20] The P4.org API Working Group, "P4Runtime Specification," Dec. 2020. [Online]. Available: https://p4lang.github.io/p4runtime/spec/v1.3.0/P4Runtime-Spec.html