SplitArchitecture

Applying the Software Defined Networking concept to carrier networks

Mario Kind, Fritz-Joachim Westphal, Andreas Gladisch, Steffen Topp
Deutsche Telekom Innovation Laboratories
Deutsche Telekom AG
Berlin, Germany
{mario.kind; fritz-joachim.westphal; andreas.gladisch; steffen.topp}@telekom.de

Abstract— **Software Defined Networking (SDN) and OpenFlow in particular have a lot of interest in the research and the academic communities. The SDN concept evolves current design philosophies and could be assumed as one approach of a future clean slate internet design. Nonetheless, the concept requires some advancement for integration in a carrier network environment. The ICT project SPARC, funded by the European Commission, is developing a SplitArchitecture concept in order to close this gap. Based on typical use cases, requirements have been derived and integrated in the process of developing the concept. The service creation for residential customers in access/ aggregation networks is a first result. Moreover an overview on the improvements by SplitArchitecture for carrier networks is provided.**

Keywords: Software Defined Networking (SDN), SPARC, SplitArchitecture, OpenFlow, use case, access/aggregation network, service creation, Broadband Remote Access Server (BRAS)

I. INTRODUCTION

In the last couple of years the importance and acceptance of telecommunication networks, especially the Internet, grows extremely fast. The Internet is already influencing almost every aspect of our life. However, the current architecture is still based on old and partly outdated design principles [1]. An improved architecture has to reflect manifold new challenges, like mobility, security, scalability etc. It is widely accepted that these challenges are not addressed by the original design principles.

At the same time the complexity and diversity of today's telecommunication networks increased significantly [1], [2]. While evolving the current architectures, complexity continues to increase by adding required functions and mechanisms (e.g. VLAN, etc.). For example, carriers look currently in two evolutions of access/aggregation networks, an improved Ethernet solution (IEEE 802.1Qad/802.1Qah [3]) and MPLS based solutions. The base technologies as such are already available since a couple of years ([4], [5]), but there are still open issues. Efforts for standardization of network architectures exist, too, for example TR-101 [6], which in practice suffers in scalability, or the evolutionary approach with the WT-145, which is not yet available (first contribution in April 2006 [7]). Overall, the number of available standards and recommendations is huge and targets the different aspects of network design and planning, but contradictory the concrete solution might be carrier dependent. For example, the bit stream access standard for Ethernet-based networks in Germany [8] suggests the use of IEEE802.1Q as 1:N model as described by TR-101 but leaves open a number of aspects (e.g. multicast) which might be selected by each carrier individually and complicates the interaction between carriers. Therefore devices and management approaches are not that uniform as the standardization efforts suggest. With each additional technology, network feature or technology migration (cf. design principle according to [9]), the complexity in the network will increase. Carrier networks do not collapse because of the fact that network administrators (still) have the ability to master complexity. But increasing complexity implicates that there is certainly room for improvement with respect to the existing network architectures. Beside the already detailed disadvantage of continuous technology changes, one of the serious disadvantages of current approaches, detailed in a following section of the paper, is the low level of abstraction [2]. Reduction in complexity in current architectures definitely demands a higher level of abstraction. Abstraction does not remove all complexity from the network but gives the opportunity to encapsulate or hide a certain degree of complexity. Furthermore abstraction is one of essential condition for a modular design and provides flexibility to perform processing in different parts of the network.

Another crucial issue is that networks are partly based on proprietary solutions where vendors include their own, not (completely) standardized mechanisms. Current network elements hide their internal flexibility. And of course the degree of flexibility differs from vendor to vendor [10]. In addition, this implies that the access to the already implemented flexibility is almost impossible. Moreover it is a serious barrier for the implementation of innovations [11]. In addition ISP are extremely restricted to organize and structure their networks in the way they prefer.

This paper gives an overview of a new network architecture approach and concentrates on the concepts of Software Defined Networking (SDN), OpenFlow protocol (OF) and the required modifications for carrier grade abilities, the SplitArchitecture. Especially the latter is subject to work conducted by the authors. SDN describes an innovative network architecture concept which is based on a clean-slate approach [1], [2]. With SDN it is possible to address various persistent problems in the area of networking by enhancing the split between control and data plane, like the already mentioned network functions for mobility, for security or the lack of an overarching design principle. In addition it should be possible to slice and virtualize the underlying network in a straightforward manner. Another design principle is the higher level of abstraction of SDN, important to increase the flexibility and reduce the complexity in network management approaches.

One of the potential main components of the SDN concept is OF. OF is an interface between control and data plane. So far OF has been used mainly in the academic area and lacks in carrier grade requirements. To overcome this demand the Open Networking Foundation (ONF) [12] has been founded. In addition to the improvement of the OpenFlow protocol itself, the target is to develop additional parts of the SDN concept like abstraction layers for network functions and application services.

Even before the demand for carrier grade improvement of the OpenFlow protocol was tackled by the ONF, a research project dealing with SplitArchitecture (ICT SPARC [13]) was started, partly funded by the European Commission in the FP7 framework. We are trying to improve the OpenFlow protocol by several aspects like carrier grade protocol support (MPLS and extensions for inter-domain compatibilities, enhanced forwarding and processing functions as well as development of typically network functions demanded for carrier grade support.

This paper is organized as follows. The second section explains the further developments of the SDN concept focused on the split architecture approaches on carrier grade networks. This is required in order to understand the work done within SPARC [13], dealing with a further separation between forwarding and processing work and extensions for reaching carrier grade level. Examples of use cases are presented and briefly explained. In the fourth section a service creation paradigm is discussed. Finally a short summary is given in the fifth section.

II. EXTENDING THE SDN CONCEPT WITH SPLITARCHITECTURE

This section starts with an overview on the SDN concept and the nexus with OpenFlow. Based on this general overview, the extensions defined by the ICT project SPARC are presented, describing the SplitArchitecture concept. It should be noted that SPARC concentrates on the use case areas of access/aggregation domain and multi-layer approach within a classical carrier grade telecommunication network (more details are provided in section IV). However, the Split-Architecture concept is generally applicable here (as well as in other areas like research facilities or data center networks). Moreover, the relationship with requirements derived from the carrier-grade network approach will be addressed.

A. The SDN concept

The notion of layering has a long tradition and was originally accepted as one of the fundamentals of network design by the ITU. According to [2], abstractions of the data plane (e.g. TCP or IP or Ethernet) are used in order to reduce complexity, but the concern is also raised that the interfaces are rather awful implemented and violate the principles of modularity. In addition, the control plane provides no abstractions at all and this results in increased complexity. Last but not least, a sophisticated management/control plane building block is demanded as well. This results in three major concerns:

- Constrained forwarding model,
- Distributed state and,
- Detailed configuration.

Based on the analysis, it is accepted that the mechanisms to overcome these concerns exist. Figure 1 a) shows todays generalized access/aggregation network structure. Control plane and data plane are integrated in the devices. However the SDN concept has to introduce three fundamental splits to introduce abstraction and therefore to overcome the issues caused by the existing complexity. These three splits are shown in Figure 1 b).

The first split separates data and control plane and moves any control functions into a Network Operating System (NOS). The second split is between NOS and control program(s) and provides a network view. This network view is further refined in order to reduce the global from the meaningful restricted view: the third split introduces a network hypervisor called nypervisor [2] between the control programs and the NOS. The corresponding network views are global network view between the NOS and the nypervisor and abstract network view

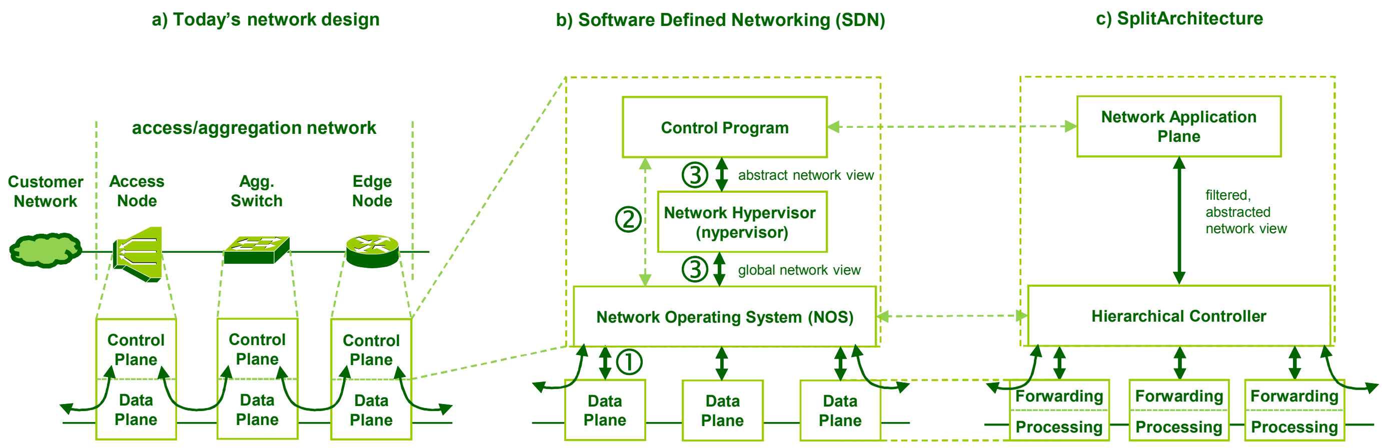

Figure 1. Comparison of today's access/ aggregation network with Software Defined Networking (SDN) and SplitArchitecture

between nypervisor and control program(s).

With SDN it is possible to address various persistent problems in the area of networking, like the already mentioned network functions for mobility, for security or the lack of an overarching design principle. Key principle of SDN is the split between control and data plane which introduces flexibility. Another important design principle is the higher level of abstraction of SDN which can be used to increase the effectiveness and to reduce the complexity of network management. Due to the key attributes of SDN described above, barriers for innovation are significantly decreased. SDN might also lower the market entry barrier for new vendors and facilitates the work of academics. The concept gives the chance to restore the control and therefore the flexibility in the network operation for ISP.

B. SDN and OpenFlow

SDN defines a general architecture only and tries to develop an alternative view on the data plane with the option to collapse layers. This means that each layer is not supposed to act separately with matching, analyzing and processing, isolated from the other layers. The SDN implementation applies OpenFlow (OF) to provide a generic matching of the layer 2 up to layer 4, lookup of actions on matches and integrated pipeline processing of the different layers. In addition OF is desired to be an implementation of a dedicated part of the general SDN concept, the connections between NOS and the network elements. In other terms, OF describes the interface protocol between the control- and data-plane or more specific – between a network controller and a forwarding interface of the data plane. OF is responsible for the exchange of information and assignments among the controller and switch.

Up to now OF is mainly evaluated by academics. Based on the fact that OF is also promising concept for commercial network operators and data centers, the Open Networking Foundation (ONF) was founded. ONF is a nonprofit organization which assesses the operational capability of OpenFlow for the deployment at large scale carrier as well as data center networks. Members of the board of ONF are: Deutsche Telekom, Facebook, Google, Microsoft Verizon and Yahoo! Meanwhile ONF associated in the order of 50 members from the different areas of the industry.

The interface description for the OpenFlow protocol specification is available in the latest version as v1.1.0 [14]. With OF the controller (part of the NOS in the nomenclature of SDN) can add, modify and delete entries in the forwarding table of a network element. On the other side, if a network element cannot find a valid entry in the forwarding table for an incoming packet, the packet will be forwarded to the controller using OF and thereby ask for instructions. The controller analyzes the packet and decides what happens with this packet or packets with similar characteristics and thereby associated to the same pattern or the same flow. This decision logic is performed or supported by control programs associated with the OF controller. The packet and the new forwarding decision will be sent back to the network element. After the reply of the controller, the network element modifies the internal forwarding table.

The latest version of the OF specification supports different data plane protocols, like Ethernet, IPv4 and MPLS with related data plane headers. In addition, it supports marking of QoS bits in Ethernet, IP or MPLS headers or modification of additional headers like VLAN. While further developments of OF will integrate more existing technologies, it potentially will enable new protocols and network functions as well. Hypothetically, an IP version beyond IPv6, next generation MPLS or Carrier Ethernet 2.0 could be possible, even with existing hardware.

C. SplitArchitecture extensions for SDN

The target of this paper is to present on both the development of architecture (the SplitArchitecture (SPARC)) and the implementation of the identified and required extensions to proof-of-concept the defined architecture. This focus of the project results in a network architecture concept with several detailed options for implementation in contrast of the very general concept reflected by SDN.

In general, SplitArchitecture acknowledges the principles of SDN with a split between data and control plane as well as the introduction of a kind of NOS. A graphical representation of the SplitArchitecture is shown in Figure 1 c). There are two substantial differences.

First the split between control program and NOS with the help of the nypervisor and two different abstractions is reduced to one abstraction, the filtered, abstract network view. This means that the complete set of functions of a network hypervisor is not mandatory. But a basic set of filter function can be used to establish a meaningful abstraction as well. Currently, the discussed design follows a pragmatic way of hierarchical controllers separating the network view by filtering the available addressing scheme and therefore granting access for control programs to parts of the resources only.

The second difference is an additional split in the data plane between forwarding and processing entities. This is motivated by different aspects. Forwarding decisions are done most efficient at the edge of the networks, but the network elements at the edge lacks of sophisticated processing capabilities in many cases (e.g. a "primitive" DSLAMs as edge device of a carrier domain in a telecommunication network). Processing capabilities are spread all over the network environment today and could be even more integrated in network devices in the future. Compared with other technologies, processing capabilities are evolving fast, so a separation of related entities could ease innovation. In addition new protocols could be integrated in existing environments more easily by the support of general purpose computing hardware for example. Moreover, at the end of the life cycle, legacy protocols could be phased out by moving the desired processing capabilities to other locations while keeping forwarding decisions still at the same network device. Referring to the access/aggregation domain (details will be provided in section IV), an example could be the PPPoE recognition in a DSLAM, but moving the BRAS functions from dedicated, single-purpose devices to a general computing hardware or a data center. Another example would be support of POTS. The responsible ingress port at the access node encapsulates everything into e.g. an Ethernet framing format (incl. TDM transition) and is forwarding to an appropriate application function responsible for analyzing the call setup and termination as well as transcoding in appropriate formats. This application function could reside in a data center, a special switch (typically called soft switch) or directly in a

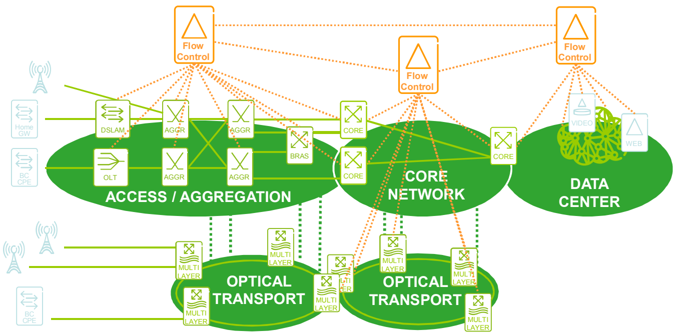

Figure 2. Overview on use case areas of SplitArchitecture

device along the transport path. The important aspect is that the access node decides on forwarding, but does not necessarily do the processing as today's concepts assume [15].

D. Requirements of a SplitArchitecture

The obvious usage of SplitArchitecture is the application in a carrier network environment; therefore carrier-grade characteristics are mandatory. Unfortunately there is no precise and complete common definition of the term "carrier-grade", so in the following a number of major aspects, which should be taken into account in the design and implementation of carrier network architecture, are listed [16] (cf. list of requirements in TR-144 [17]).

- Capabilities to provide multi-service / -provider operation must be ensured. Especially in the context of a shared infrastructure, aspects like different address schemes or dependencies of administrative domains have to be taken into account. It is also important to support interconnection to other domains, including the support of diverse OAM mechanisms, exchange of authentication information and control plane protocols for routing.
- Mobile backhaul should be supported by appropriate mechanisms like TDM emulation.
- Multi-protocol support at different layers should be supported. This addresses the parallel operation of data plane protocols like Ethernet in various options, MPLS or different versions of IP.
- Authentication, authorization and accounting (AAA) in conjunction with the distribution of customer profiles and related enforcement in network devices needs to be supported. This is an important part of the service creation concept and thereby the network access of end-customers. This is one of the fundamental differences to a LAN environment.
- Best practices of Quality of Service (QoS) should be supported in order to enable the forwarding guarantees required by the services.
- OAM must be supported for the analysis of failures and misconfiguration issues in the network. Most data plane protocols have already implemented OAM mechanisms and it should be checked whether those mechanisms could be coordinated or reused.
- Sufficient network management is mandatory while network devices might be located in quit a distance from the next network service team. Again, best practice principles and mechanisms exist and should be reused or revised.
- Security and protection of infrastructure/architecture components must be taken into account. This should include manifold aspects, like control of broadcast domains, the enforcement of traffic forwarding principles, control of identifiers in order to prevent spoofing attacks as well as the general limitation of the number of identifiers in order to avoid overflows.
- Multicast to support traditional television services has to be considered.
- Data plane operation should support chains of processing in order to assist complex processing.
- Support of legacy solutions as well as openness for future protocols should be considered.
- Finally, circuit switching needs to be taken into account as well in the design of the SplitArchitecture.

III. SPLITARCHITECTURE USE CASES

Because of the fact that SplitArchitecture ensures an enhanced grade of flexibility and modularity it presents a wide spectrum of alternative use cases. Studying potential areas of implementation of a SplitArchitecture approach in carrier grade networks brought up that the most important use case areas are the access/aggregation domain, the data center domain and multi-layer approaches. The ICT project SPARC especially concentrates on the access/aggregation part inclusive the mobile backhaul. The rest of the paper will concentrate on the access/aggregation uses cases. The other use case areas are not covered in detail. In Figure 2 the generic use case areas are depicted in a simplified manner.

Even the access/aggregation domain covers a lot of different opportunities to implement OF. The paper concentrates on the following:

- Multi-service/multi-provider operation
- Mobile backhaul
- Dynamic control composition

One requirement that is driving the evolution of the access/ aggregation domain is the imperative to increase available bandwidth. Furthermore carriers are engaged to enable access for competitors on the access/aggregation network environment. A constraint of shared use of infrastructure is to guarantee the total separation of access or influence of the control planes of the different operators to each other. By using OF, the isolation of different slices can be done by means of setting up a virtualized physical infrastructure on top. In addition, OpenFlow supports tag-based differentiation in order to isolate traffic of different providers. These two opportunities to separate the slices of different providers may be used in multi-provider scenarios of OF-enabled access/aggregation domains.

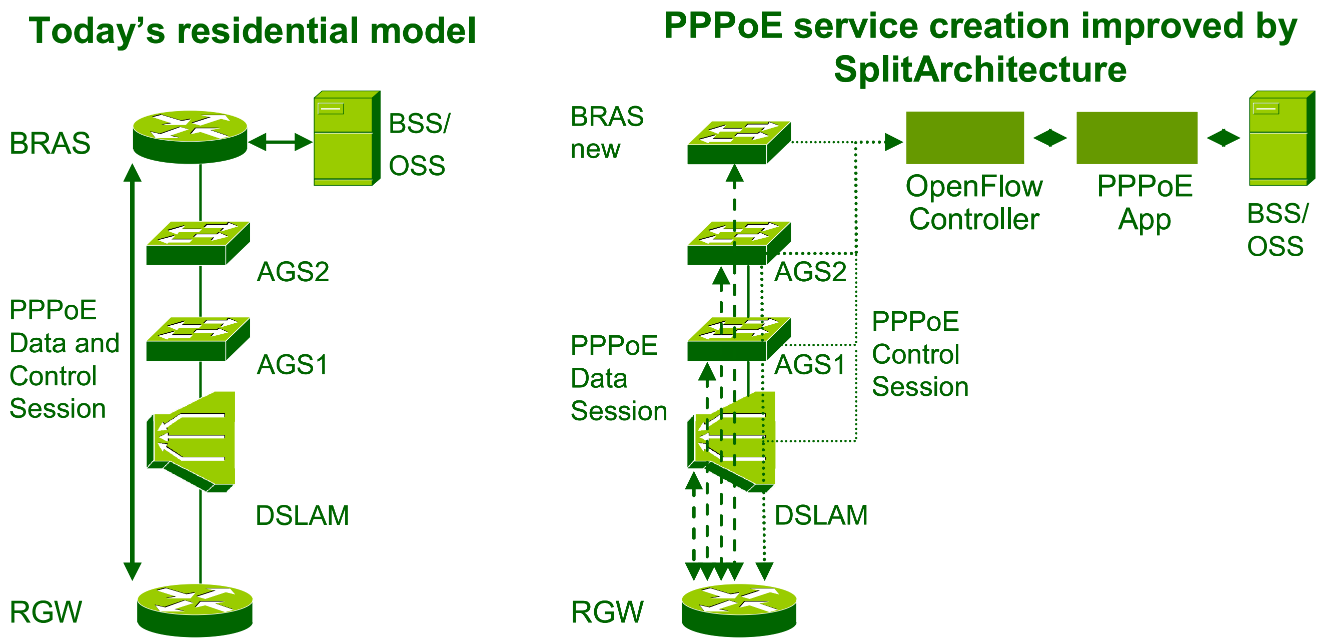

Figure 3. Service creation with today's residential model and PPPoE service creation improved by SplitArchitecture

Usage scenarios are shifting more and more from wired to wireless mobile access. OF can also be used to improve the interior functionality of the mobile backhaul network. Most aspects of mobile backhaul are covered by multi-service/multi-provider use cases. However OF applies a high capacity packet transport service between the mobile base station and the serving gateway using the S1 interface. Another important aspect is that OF enables the definition of common network functions regarding QoS, AAA and policy control. This has a profound impact on fixed mobile convergence concepts.

IV. SplitArchitecture Example for Service Creation

This section provides an overview and an example of the use case of service creation in a carrier environment. Due to two reasons it is not possible to present all details. First, it is ongoing work in the project SPARC, second there are a reasonable large number of potential options and variations. So this section is limited to present the general idea of service creation and highlight the impact and potential improvements with SplitArchitecture. Other already ongoing work is the improvement of protocol support for MPLS (see [18] or [19]) and a scalability analysis.

A. Service Creation

Carrier networks differ in a fundamental design principle from LAN environments: User are not trusted, change their behavior and therefore must be authenticated, configured and controlled.

The concept to enable the desired control and configuration is the key of the service creation. In general, the idea is to define entities in the network where a service is defined and configured in order to reduce the complexity of operation. These central entities refer to the configuration of services with customer relationship and do not include transport related functions (like, forwarding packets only without any desired customer relationship, e.g. encapsulation of traffic of a DSLAM into a MPLS pseudowire). Typically, service creation is done in edge devices.

Today, two models exist. The first and more common one is based on the Point-to-Point Protocol (PPP), which is rather a protocol but a protocol suite providing functions like:

- Sophisticated and cohere model
- Authentication and authorization
- Configuration on layer 2 to 7
- Session control or OAM (and to some extent control for accounting as well)
- Interfaces to OSS/BSS platforms based on e.g. RADIUS

A typical PPP implementation today is based on a Broadband Remote Access Server (BRAS) at the boarder in-between access/aggregation and core networks. The PPP over Ethernet (PPPoE) implementation is linking data and control plane and the BRAS is the central session termination point in a single purpose, complex and costly device.

The second model is based on the Dynamic Host Configuration Protocol (DHCP), sometimes referred to as IP Session concept [20]. The basic function of DHCP is the configuration of network device with information for Layer 3 to 7. Therefore, additional functions or protocols need to be taken into account for a complete and sophisticated implementation. The next section concentrates on the SplitArchitecture applied with service creation based on the PPPoE model.

B. PPPoE model improved by SplitArchitecture

The previous section sketched the general basics of service creation based on PPPoE. This section will provide details about potential realizations of SplitArchitecture and the associated benefits.

Figure 3 shows on the left side today's service creation model with centralized, integrated BRAS and the access/aggregation network. Both data and control session parts are tunneled by PPPoE. The BRAS is responsible for the interaction with the OSS/BSS platforms.

On the right side of Figure 3, the split between control and data plane applied on PPPoE is shown. The control termination part from BRAS is shifted into a PPPoE app (PPPoE control session), which behaves and reacts similar to today's BRAS model and is responsible for session management and interaction with the OSS/BSS. Control plane information could be forwarded by any device in the network processing chain (DSLAM, Aggregation Switches (AGS1 & AGS2) or BRAS new), which requires only appropriate forwarding entries on the device. The data plane part remains on the devices in the original network processing chain. This includes two aspects. First the control plane information is sent to the PPPoE app as explained previously and this forwarding could be integrated in an improved version of the OpenFlow protocol. Second, the data session part of PPPoE is forwarded along the network processing chain to the core network and is therefore independent of the control information. It must be coupled with processing functions for PPPoE, which in turn require some more information and functions. The basic idea is to trigger this processing entity by the PPPoE app with the help of an extended OpenFlow protocol (other options like a separate protocol approach exist as well).

Each of the devices in the network processing chain could be potentially used for the processing tasks, but other options like external processing platforms (e.g. a PC attached to the device or general purpose computing hardware integrated in device) exists as well. This represents the second split, as proposed by the SplitArchitecture concept (see section II.C), with a division of the data plane in independent forwarding and processing entities. More details on this could be found in [21].

V. Conclusion and Next Step

This paper briefly describes the concept of Software Defined Networking and the OpenFlow protocol. The document also explains the idea of the SplitArchitecture and its difference with respect to Software Defined Networking. Based on a set of requirements to define the characteristics of carrier grade, use case areas of SplitArchitecture are discussed and use case in the access/aggregation domain of carrier networks are described in more detail.

As part of our future agenda we will investigate the mentioned use cases of SplitArchitecture in more detail and proof-of-concept the most important functions. In addition, we are trying to analysis scalability issues like performance of the hierarchical Controller. Moreover a techno-economic study will complete the ongoing study.

Acknowledgment

This work has been partially supported by the European Commission through the FP7 Project: Split Architecture (SPARC) under the grant agreement N° 258457. The authors would like to thank Dr. Hagen Wösner, Andreas Köpsel and Tobias Jungel for their support in the development of the PPPoE based service creation with SplitArchitecture. The authors also would like to thank the members of the project SPARC for their very valuable support and feedback in the development of the SplitArchitecture.

References

[1] Anja Feldmann, "Internet clean-slate design: what and why?", ACM SIGCOMM Computer Communications Review (CCR), volme 37, number 3, pages 59-64, July 2007

[2] Scott Schenker, "An attempt to motivate and clarify Software-Defined Networking (SDN)", talk at Ericsson Research, 2011, available at http://www.youtube.com/watch?v=WVs7Pc99S7w&feature=player_embedded#!

[3] Anthony Jeffree et al., "Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks", The Institute of Electrical and Electronics Engineers, Inc., ISBN 978-0-7381-6708-4.

[4] Robert M. Grow et al., "Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", The Institute of Electrical and Electronics Engineers, Inc. , ISBN 973-07381-5796-2.

[5] E. Rosen, A. Viswanathan, R. Callon, "RFC3031: Multiprotocol Label Switching Architecture", available at http://www.rfc-editor.org/rfc/rfc3031.txt

[6] Tom Anschutz et al., "TR-101 Migration to Ethernet-Based Broadband Aggregation, Issue 2", Broadband Forum, July 2011.

[7] Geraldine Calvignac, Gilles Bourdon, "MPLS-enabled Access Networks", contribution dsl2006.286.00, submitted April 26th 2006, not publicly available.

[8] NGA Forum, "L2-BSA II - Technische Spezifikation", 10.10.2011, available at http://www.bundesnetzagentur.de

[9] R. Callon, "RFC1925: The Twelve Networking Truths", available at http://www.rfc-editor.org/rfc/rfc1925.txt

[10] Nick McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks", whitepaper, 2008

[11] Temu Koponen et al., "Architecting for Innovation", ACM SIGCOMM Computer Communications Review (CCR), volume 38, number 2, 2011

[12] The Open Networking Foundation, website https://www.opennetworking.org/

[13] The SPARC Consortium, project website: http://www.fp7-sparc.eu

[14] The OpenFlow Switch Consortium, "OpenFlow Switch Specification Version 1.1.0", 2011, available at http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf

[15] Keymile, "White Paper: Advantages of Voice-over-IP with IP-based multi-service access nodes (IP MSAN)", available at http://www.keymile.com

[16] The SPARC Consortium, Deliverable 2.1 "Initial definition of use cases and carrier requirements", ICT-258457, 2011, available at http://www.fp7-sparc.eu/project/deliverables/

[17] Anna Cui et al., "TR-101 Broadband Multi-Service Architecture & Framework Requirements", Broadband Forum, August 2007.

[18] Attila Takacs et al., "Combining GMPLS and OpenFlow for Transport Network Applications", available at http://www.pilab.jp/ipop2010/info/onlineproceedings.html#1-2

[19] Howard Green, James Kempf, Sonny Thorelli, Attila Takacs, "MPLS Openflow and the Split Router Architecture: A Research Approach", http://www.isocore.com/mpls2010/program/abstracts.htm#wed1_5

[20] IST MUSE Contract No. 026442, "Part A: GSB Access Network Architecture", deliverable 1.9, 2008, available at http://www.ist-muse.org

[21] The SPARC Consortium, Deliverable 3.2 "Update on Split Architecture for Large-Scale Wide-Area Networks", available in December 2011at http://www.fp7-sparc.eu/project/deliverables