Applying Software-Defined Networking to the Telecom Domain

Georg Hampel, Moritz Steiner and Tian Bu
Bell Labs – Alcatel-Lucent
Murray Hill, New Jersey, USA
georg.hampel/moritz.steiner/tian.bu@alcatel-lucent.com

***Abstract*— The concept of Software-Defined Networking (SDN) has been successfully applied to data centers and campus networks but it has had little impact in the fixed wireline and mobile telecom domain. Although telecom networks demand fine-granular flow definition, which is one of SDN's principal strengths, the scale of these networks and their legacy infrastructure constraints considerably limit the applicability of SDN principles. Instead, telecom networks resort to tunneling solutions using a plethora of specialized gateway nodes, which create high operation cost and single points of failure. We propose extending the concept of SDN so that it can tackle the challenges of the telecom domain. We see *vertical forwarding*, i.e. programmable en- and decapsulation operations on top of IP, as one of the fundamental features to be integrated into SDN. We discuss how vertical forwarding enables flow-based policy enforcement, mobility and security by replacing specialized gateways with virtualized controllers and commoditized forwarding elements, which reduces cost while adding robustness and flexibility.**

***Index Terms*—Software-defined networking, telecom, cellular network, fixed wireline network, tunneling, gateway**.

I. Introduction

Software-Defined Networking (SDN) aims for the separation of control- and forwarding planes by substituting distributed forwarding or routing protocols with centralized control [1]. SDN's centralization of control has a variety of advantages: The controller function obtains a global picture of the network resources, can run centralized optimization algorithms and swiftly implement flow changes to the forwarding plane, circumventing the convergence problems commonly known from distributed signaling protocols. SDN further permits flow definition with fine granularity as well as load balancing among paths and forwarding elements (FEs).

SDN has been applied to networks of confined size, i.e. where the number of FEs is small (e.g. < a few hundred), such as data centers and campus networks. It has also been used to interconnect data-center networks [2]. While the network extends over a large physical scale in this latter case the total number of nodes, i.e. data centers, still falls within a range where centralization of control remains manageable.

SDN has not yet played a major role in the telecommunications domain such as wireless and fixed wireline access networks. This may seem surprising since fine-granular flow differentiation is relevant in such networks to support service- and subscriber-specific policies and charging. Another reason is mobility support which demands subscriber-specific forwarding. In the future, the trend toward fine-granular forwarding is expected to increase with the introduction of application-based networking concepts, more refined billing solutions, enterprise-based Access Point Names (APNs) and operator virtualization.

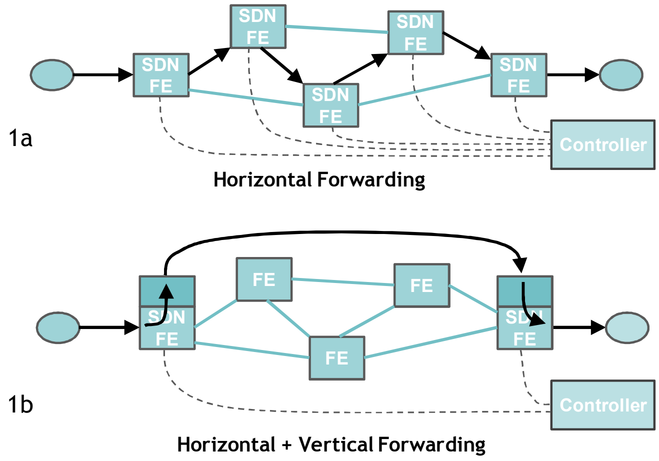

Figure 1: SDN forwarding concepts. 1a: horizontal forwarding, 1b: horizontal + vertical forwarding.

Currently, telecom carriers address these requirements via overlay networks using tunneling solutions, which permit subscriber- and service-specific flow management on top of a native IP-, MPLS- and Ethernet-based infrastructure. While the underlying infrastructure consists of many FEs using conventional routing or forwarding protocols, the tunneling end points – typically referred to as gateways – remain few in number, which limits capital expenditures as well as signaling load. Gateway solutions have therefore been applied throughout the mobile and fixed wireline telecom domain. Apart from mobility and service differentiation, gateways and tunneling have also been used for data protection, to facilitate virtual private networks, for IPv4-to-IPv6 protocol transition and to conduct traffic engineering.

While network overlays can master telecom-specific requirements, the gateway functions introduce a variety of problems. Since gateways hold flow-specific state information they become single points of failure, demanding elevated reliability requirements and therefore higher cost. Further, the many purposes for tunneling have created a myriad of signaling protocols and associated gateway functions, whose specialization defies gateway commoditization.

In addition, gateway- and vendor-specific hardware implementations have led to a plethora of specialized "boxes" each of them demanding its own maintenance procedure and supply chain of repair pieces, which drives up operation expenses. These hardware implementations have further proven inflexible to feature upgrades.

Finally, the very existence of specialized nodes has motivated standards bodies to attach even more functions to these boxes, which reinforces the above predicament.

Our Contribution: We propose to extend the concept of SDN to controller-programmed en- and decapsulation operations on top of IP. We refer to this extension as *vertical* forwarding since it permits SDN to forward data *across* networking layers such as a native IP substrate and a tunnel-based overlay. *Vertical* forwarding stands in contrast to *horizontal* forwarding, which has been the present focus of SDN and emphasizes on data forwarding *within* the same networking layer (Fig. 1). With SDN extended to vertical forwarding, we enable a new paradigm for telecom architectures that allows substituting specialized gateways with generic SDN FEs and centralized control. In this manner, the advantages of SDN are combined with those of tunneling while avoiding their respective shortcomings.

The next section discusses vertical forwarding and the associated paradigm shift for telecom architectures. In section 3, we apply the new paradigm to a variety of use cases in the telecom domain and discuss the associated advantages. In section 4, we present how vertical forwarding can be integrated into the SDN-protocol OpenFlow [3]. Section 5 discusses related work. We summarize our work in the conclusion.

II. Vertical Forwarding

Present SDN concepts focus on forwarding data within the same networking layer along a path of FEs, which we refer to as *horizontal* forwarding (Fig. 1a). In this architecture, fine-granular flow definition is *only* possible if all FEs can be programmed by the controller. This requirement fails when the number of FEs becomes too large or when a substantial fraction of FEs does not speak the SDN protocol, e.g. due to legacy reasons or because they pertain to a different administrative domain. All of these reasons apply to typical telecom networks.

We propose to extend the SDN concept to *vertical* forwarding, which facilitates data migration *between* networking layers such as from a native forwarding substrate to an overlay or between layers of stacked network overlays (Fig. 1b). Vertical forwarding requires en- and decapsulation operations on top of IP to be integrated into the FE and placed under the direction of the controller. Vertical forwarding should not be confused with some realizations of SDN, where horizontal forwarding is exercised *on top* of a network overlay, such as a VPN, and tunneling and de-tunneling operations remain transparent to the controller [2].

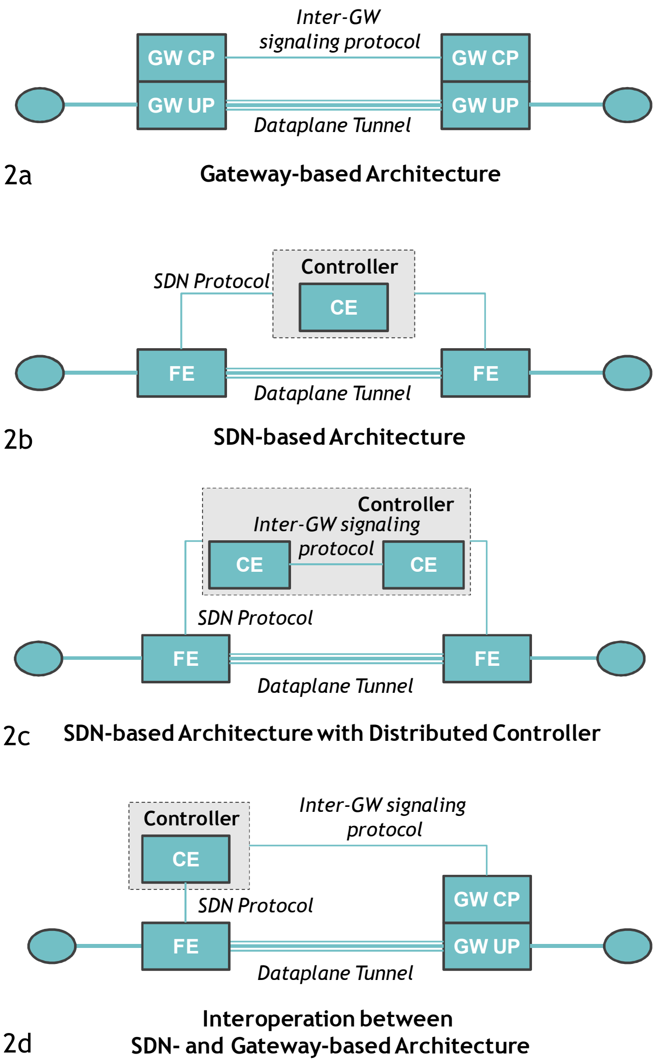

Figure 2: Telecom architectures for data tunneling. 2a: Gateway architecture; 2b: SDN-based architecture with vertical forwarding and centralized controller; 2c: same with distributed controller; 2d: mixed architecture with one SDN-controlled FE and one gateway

Vertical forwarding permits the controller to create flows with fine granularity while using only a few controllable FEs, referred to as *SDN FEs*, which tunnel and de-tunnel the user-plane data. Their small number keeps centralized control within scalability bounds. At the same time, the network's remaining infrastructure can apply conventional routing protocols dictated by legacy and administrative domains.

The number of tunneling protocols SDN FEs have to support is small compared to the myriad of signaling protocols used for control plane operations. The typical layer-3 encapsulation protocols are IP, UDP, GRE, GTP, a few IPv6 extensions, ESP and AH. Also Ethernet (ETH) and PPP are used on top of IP even though they represent native layer-2 protocols. Since many en- and decapsulation operations are rather similar it can be anticipated that future SDN FEs will support them all. Consequently, these FEs can be flexibly applied for all tunneling purposes.

With tunneling integrated into SDN, it is now possible to apply SDN as a remedy to telecom gateways. For that purpose, each gateway is split into control- and data-plane section, where the former is integrated into the controller while the latter is supported by an SDN FE. The controller hence becomes the specialized entity that buries the myriad of telecom signaling protocols in support of user- and service-specific policy enforcement, charging, mobility and security. The controller may be realized as a distributed entity that is vertically split into a network operating system and management layer or horizontally split into multiple instantiations to improve scalability.

SDN, when extended to vertical forwarding, permits a paradigm shift in the design of telecom architectures. This shift has a variety of advantages.

Firstly, implementations of controller and FE can be tailored to the respective requirements these functions bear. FEs can be optimized for high throughput and eventually become commoditized due to their small and (rather) stable feature set. The controller functions can be virtualized and leverage general purpose hardware, which reduces operation costs and enables more cost-effective methods to provide reliability [4]. Further, network operators can conduct infrastructure upgrades, e.g., to increase capacity, independently for forwarding and control plane.

The separation of the control plane from the forwarding infrastructure also creates higher flexibility to standards and feature upgrades. Since such upgrades are mostly limited to the control plane they can be easily applied to software-based controller solutions without affecting commoditized FEs.

SDN-based architectures further permit more flexible network deployments than gateway-based architectures. The latter generally dictate specific gateway arrangements as well as the associated inter-gateway signaling protocols (Fig. 2a). The SDN-architecture supports a variety of scenarios in contrast: In one solution, all control plane functions are supported by one centralized controller (Fig. 2b), which omits all inter-gateway signaling protocols and only dataplane tunnels remain in place. In addition, an SDN protocol is run between controller and FEs.

It is further possible to distribute the controller function over multiple nodes, where the various controller instantiations mutually communicate via the original inter-gateway signaling protocol or via a proprietary protocol (Fig. 2c).

The controller can also interoperate with a legacy gateway solution in case one tunnel end point is held on an SDN FE and the other on a gateway. In this scenario, the controller programs the SDN FE via SDN protocol and interacts with the gateway via the standards-specific signaling protocol (Fig. 2d). This flexibility permits incremental deployment of an SDN-based solution as well as optimizations to be conducted on the signaling plane.

Finally, the SDN-based architecture provides improved robustness against link or node failure (Fig. 3). Figure 3b shows a scenario of a data flow established between two hosts A and B, which is tunneled between FE1 and FE2a. The FEs could represent a Mobility Access Gateway (MAG) and a Local Mobility Anchor (LMA) in Proxy Mobile IP (PMIP) [5], for instance. In case FE2a or a link to FE2a fails, any conventional routing protocol will direct data from host B along an alternative path to FE2b. The controller can then enter a replica of FE2a's downlink flow entries onto FE2b and also redirect uplink flows on FE1 toward FE2b. In this manner, end-to-end connectivity is reestablished.

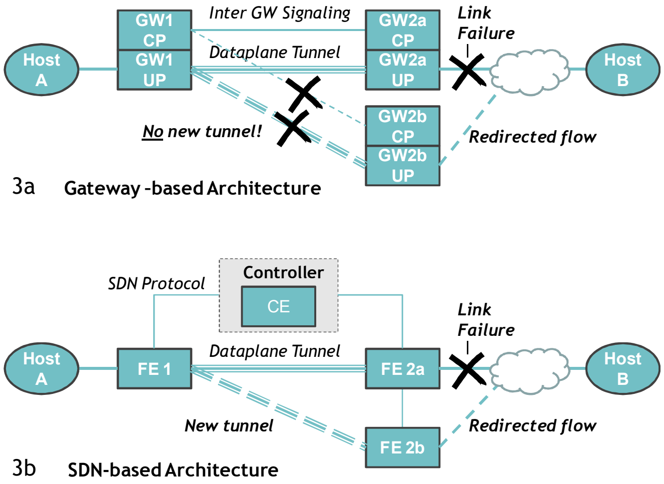

Figure 3: Robustness to link failure: Gateway architecture (3a) vs. SDN-based architecture with vertical forwarding (3b).

An equivalent redirection is not possible in the gateway-based architecture since each gateway hosts its own control-plane function (Fig. 3a). Therefore, GW1 and GW2a sustain a tight one-to-one correspondence, which does not permit GW1 to redirect its flows to GW2b since the latter does not hold the associated state information.

One might argue that the improvement in robustness critically depends on the reliability of the controller. This reliability, however, can be easily and inexpensively accomplished by implementing this function as a virtualized software solution, which can be synchronized with backup facilities and migrated upon hardware failure.

III. USE CASES

We discuss the applicability of the SDN-based architecture to the telecom domain on hand of a predominant set of use cases. Since these use cases apply to the fixed wireline as well as the mobile domain the SDN-based architecture becomes an enabler for fixed-mobile convergence on networking layer.

A. IETF Mobility Protocols

Since IP does not natively support host mobility, the IETF developed a variety of mobility protocols that run on top of IP. Many of these protocols have been adopted by the mobile telecom standard bodies 3GPP and 3GPP2.

Mobile IPv4 (MIPv4) introduces two gateway functions referred to as Home Agent (HA) and Foreign Agent (FA), which represent a global and a local anchor function, respectively [6]. The signaling plane consists of Registration Request and Reply messages exchanged between mobile nodes (MN) and the two gateways. On the data plane, an MN-specific

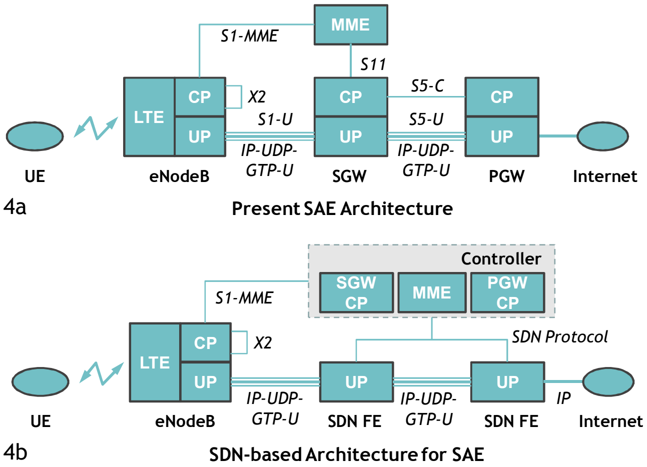

Figure 4: SAE architecture using gateways (4a) and proposed SDN alternative using vertical forwarding (4b).

tunnel is established between HA and FA using IP-in-IP encapsulation [7] or Generic Routing Encapsulation (GRE) [8]. The former adds an IPv4 header and the latter an IP-GRE header stack to the packet. MIPv4 has been adopted by 3GPP2 for 3G mobile networks even though in slightly varied form. In the SDN-based architecture, the MN exchanges the registration messages directly with the controller. Global and local anchors are represented by SDN FEs that sustain MN-specific IP-in-IP or GRE tunnels.

Mobile IPv6 (MIPv6) also uses the HA as a global anchor but it moves the local anchor into the MN. Signaling is conducted between MN and HA via binding update messages [9]. On the user plane, MIPv6 applies generic tunneling in IPv6, i.e., IPv6-in-IPv6 encapsulation [10]. MIPv6 is currently used by 3GPP to support mobility to untrusted domains [11]. In the SDN-based architecture, MN and controller exchange binding updates while only one SDN FE is required on behalf of the global anchor.

Proxy Mobile IPv6 (PMIP) re-introduces the local anchor referred to as Mobile Access Gateway (MAG)[5]. The global anchor function is (confusingly) referred to as Local Mobility Anchor (LMA). Opposed to MIPv4 and MIPv6, the mobile node is not involved in PMIP-related signaling. PMIP also applies generic packet tunneling in IPv6. PMIP is currently used as alternative mobility solution in 3GPP's System Architecture Evolution (SAE). In the SDN architecture, PMIP's binding update messages remain inside the controller domain. On the dataplane, MAG and LMA are replaced via SDN FEs that sustain generic packet tunneling.

B. 3GPP UMTS & SAE

3GPP supports 3G and 4G mobile network architectures referred to as Universal Mobile Terrestrial System (UMTS) and System Architecture Evolution (SAE), respectively, which pursue mobility support, service differentiation and operator virtualization [12]. Both architectures support three IP-aware hierarchy layers.

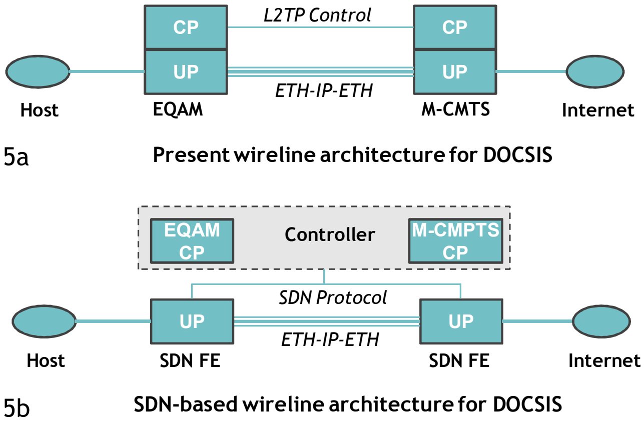

Figure 5: Architecture typically used in fixed wireline domain for cable access (5a) and proposed SDN alternative using vertical forwarding (5b).

We mainly focus on SAE which represents the most recent fully IP-aware 3GPP standards development. The gateways associated with the three hierarchy layers are referred to as eNodeB, Service GateWay (SGW) and Packet-data-node GateWay (PGW) (Fig. 4a). While the eNodeB provides wireless access, the SGW can be interpreted as a local mobility anchor and the PGW as an enforcement point for service differentiation. The signaling between eNodeB and SGW runs via the Mobility Management Entity (MME), which represents a plain control plane node. The corresponding signaling interfaces are referred to as S1-MME (between eNodeB and MME), S11 (between MME and SGW) and S5-C (between SGW and PGW). Userplane data are bundled to EPS bearers, which create subscriber-, service- and APN-specific traffic differentiation, and tunneled along the eNodeB-SGW-PGW chain via the concatenated interfaces S1-U and S5-U. The tunnels use GTP encapsulation which inserts an IP-UDP-GTPU header stack into each packet.

When migrating SAE to an SDN-based architecture, the control plane functions of SGW, PGW and MME can be comprised into the controller, while data forwarding is conducted by SDN FEs applying GTP (Fig. 4b). Based on deployment scenario, either one or two FEs can be employed. While the eNodeB also holds a gateway function, which could be integrated into the SDN architecture, the associated benefit is small since this node remains highly specialized due to its wireless LTE interface and the associated signaling exchange with the core network (via S1-MME) and other eNodeBs (via X2 interface). Further, eNodeBs are deployed in large quantities to provide area coverage, which already leads to some degree of commoditization.

The same architecture transformation can be applied to UMTS. The corresponding gateway functions are RNC, SGSN and GGSN with respective interfaces IuPS (between RNC & SGSN) and Gn (between SGSN & GGSN). These interfaces carry signaling as well as user-plane traffic differentiated through respective GTP-C and GTP-U encapsulation.

C. Wireline Broadband Networks

Wireline broadband networks can be provided via twisted pair copper, cable or optical fiber. The common access standards are referred to as Digital Subscriber Line (DSL), Data Over Cable Service Interface Specification (DOCSIS) or FTTX (Fiber to the X) [13]. In all of these access solutions, the trend moves toward user- and service-specific flow-definition at a central node to provide appropriate policy, QoS and charging profiles.

For DSL, policy enforcement is provided by the Broadband Remote Access Server (BRAS) which interconnects with a DSLAM at the network edge via per-flow PPP tunnels on top of an Ethernet substrate (PPPoE)[14].

The equivalent central policy enforcement point for DOCSIS is referred to as M-CMTS, which interconnects with an EQAM node at the edge (Fig. 5a). Flow differentiation between both gateways is supported via L2TP pseudo-wires (encapsulation with IP-ETH header stack) [15].

In the SDN-based architecture, the discovery stage of PPPoE or L2TP control messages resides inside the controller while PPPoE or IP-ETH payload encapsulation is conducted by appropriate SDN FEs (Fig. 5b).

D. Virtual Private Networks and Secured Links

Virtual Private Networks (VPNs) are used to interconnect individual hosts with a security gateway, e.g., an off-site employee with its enterprise, which requires dynamic and fine-granular flow definition. For this purpose, an IPsec tunnel consistent of IP-UDP-ESP- or IP-UDP-AH header stack is established between the security gateway and the remote host [16]. ESP [17] and AH [18] serve security purposes, i.e. encrypt (ESP) and/or authenticate (ESP/AH) the payload. The security associations (SAs) between both gateways are established via signaling protocols such as the Internet Key Exchange (IKE2) [19]. Another example applies to 3GPP's solution for subscriber mobility to untrusted domains, e.g., from an LTE network to a Wi-Fi hotspot. In this scenario, an IPsec tunnel is established between the evolved Packet Data Gateway (ePDG) on the 3GPP network and the User Equipment (UE) [11].

When applying the SDN architecture to such VPNs, the signaling protocol, e.g. IKE2, falls into the realm of the controller. In case one end point represents a single host, this host sustains signaling with the controller. The SDN FE further has to conduct payload en- and decryption (ESP) and compute or verify packet authenticators (ESP, AH). For this purpose, the controller has to forward the appropriate keying material to the SDN FEs via the SDN protocol.

E. IP Protocol Transition

During IPv4-to-IPv6 transition, relay gateways are employed to ensure IPv6 connectivity across IPv4 domains via tunnels that encapsulate IPv6 packets in IPv4. There are two IP-in-IP encapsulations referred to as 6to4 [20] and 6in4 [21]. Teredo is another tunneling mechanism which uses IPv4-UDP encapsulation permitting traversal of middle boxes such as firewalls in the IPv4 domain [22]. These tunneling solutions currently require external means of configuration at the tunnel end points. With SDN extended to vertical forwarding, the controller can orchestrate the appropriate configuration.

IV. Realization with OpenFlow

Vertical forwarding can be embedded into existing SDN solutions such as OpenFlow [3] or FORCES [23]. In the following, we discuss the necessary changes for OpenFlow, which affect the OpenFlow protocol as well as pipeline processing on the OpenFlow FE.

For the OpenFlow protocol, additional *flow match types* and the corresponding *flow match fields* have to be introduced for parameters held on encapsulation headers. Examples are TEID for GTP-U, KEY for GRE and SPI for ESP/AH, etc.

The OpenFlow protocol further requires extensions to enable en- and decapsulation operations, which can be represented as *actions* that are applied after flow matching. For this purpose, the *push/pop actions* OpenFlow 1.3 already supports for VLAN tags and MPLS labels can be leveraged (i.e. pushIPv4, popUDP, pushGRE, etc). Each *push action* is followed by the corresponding *set_field* actions, which allow the controller to supply the necessary header field values.

When pushing or popping ESP and AH, information on the flow's security policies and security associations have to be provided. Some of this information is already supplied by OpenFlow through different means. Traffic selection as defined by IPsec [16], for instance, is captured via OpenFlow's flow-matching operations. To manage the lifetime of SAs, the lifetime of OpenFlow's table entries can be leveraged. ESP/AH push and pop actions still have to contain fields related to encryption and authentication algorithms and keying material.

Extending OpenFlow to vertical forwarding imposes additional operations on the FE pipeline processing. Firstly, the FE has to characterize the header stack of an incoming packet by using the IANA-defined protocol- or next-header field entries or the UDP destination port numbers for UDP-encapsulated headers. This has to be done prior to flow matching so that it can be used to characterize match fields on the various headers.

After flow matching, all pop actions associated with this table entry are executed first. When popping ESP or AH headers, the respective payload decryption and authentication operations are conducted. The header stack analysis may have to be continued after decryption. Then, all push- and set-field actions are executed. Finally, all operations involving multiple headers or payload, such as the population of "next header" fields, computation of transport-layer checksums and authenticators, or ESP-based payload encryption, are exercised. The packet should have a consistent header stack before it is passed to the next table or the output.

ESP, AH and sometimes also GRE and GTP-U headers contain sequence numbers (SNs) for security reasons. When pushing or popping these headers, the OpenFlow FE has to compute an appropriate SN or verify the SN contained in the header. For this purpose, SNs used on prior packets have to be buffered in the table entry of each flow. Holding flow-specific state on the FE is not a principal novelty for OpenFlow since such a feature is already in place for metering and flow statistics.

Finally, OpenFlow FEs performing vertical forwarding ought to interoperate in a legacy infrastructure using IP together with conventional routing protocols. For this purpose, OpenFlow has defined the concept of the *virtual port*, which hides IP routing protocols from the FE's SDN-related activity. In our use cases, little interference is to be expected between OpenFlow-based forwarding and the routing function behind the virtual port since prior focuses on vertical and the latter on horizontal forwarding. It is also possible to integrate distributed routing protocols into OpenFlow's controller operation.

V. Related Work

A variety of attempts has been made to simplify or improve present telecom architectures, mainly in the mobility domain.

One publication proposes the virtualization of the evolved packet core (EPC) to offer mobility as a service [24]. While this effort benefits control-plane nodes, such as the MME, gateway virtualization is expected to limit dataplane performance. Further, the principal shortcomings of gateway-based architectures discussed above cannot be addressed.

OpenFlow 1.3 introduced implicit support for GRE tunneling. It allows the controller to specify a *tunnel id*, representing the GRE KEY, and pass it to a *virtual port* where GRE encapsulation is exercised. A similar solution was proposed for GTP [25]. While this procedure permits attaching GRE or GTP encapsulation to an OpenFlow FE it does not integrate vertical forwarding into SDN and cannot be extended to a larger number of header encapsulations. An OpenFlow FE with tunneling line card therefore remains a specialized, vendor-specific hardware node.

The DMM working group in the IETF aims to define a new network architecture for mobility support, which reduces the need for tunneling by pushing anchor functionality further to the edge [26]. This approach is beneficial when mobility is sought but it does not extend to telecom requirements such as service-differentiation, operator virtualization and security.

VI. Conclusion

We presented an architecture evolution of telecom networks toward an SDN-centric paradigm. As discussed, such a paradigm shift has profound advantages to network operators such as lower operation cost, increased flexibility and ultimately higher performance.

While we see great potential for this paradigm shift to happen more discussions are needed within the research community and telecom industry. Further, efforts to define and standardize an appropriate SDN protocol with support for vertical forwarding would be desirable.

References

[1] ONF Market Education Committee, "Software-Defined Networking: The New Norm for Networks", http://www.opennetworking.org, April, 2012.

[2] Urs Hoelzle, "The Google OpenFlow network is in Production", Open Network Summit, April 2012.

[3] http://www.openflow.com

[4] M. F. Mergen, V. Uhlig, O. Krieger and J. Xenidis "Virtualization for high-performance computing", ACM SIGOPS Operating Systems Review, Vol. 40, April 2006

[5] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury and B. Patil, "Proxy Mobile IPv6", RFC5213, IETF, Aug. 2008

[6] C. Perkins, "IP Mobility Support for IPv4", RFC3344, IETF, Aug. 2002

[7] C. Perkins, "IP encapsulation with IP", RFC2003, IETF, Oct. 1996

[8] S. Hanks, S, T. Li, D. Farinacci and P. Traina, "Generic Routing Encapsulation (GRE)", RFC1701, IETF, Oct 1994

[9] C. Perkins, D. Johnson & J. Arkko, "Mobility Support in IPv6", RFC6275, IETF, July 2011

[10] A. Conta & S. Deering, "Generic Tunneling in IPv6", RFC2473, IETF, Dec. 1998

[11] 3GPP, "Architecture enhancements for non-3GPP accesses (Release 11)", TS 23.402

[12] E. Dahlman, S. Parkvall, J. Sköld, "4G: LTE/LTE-Advanced for Mobile Broadband", Academic Press, Elsevier, Oxford, UK, 2011

[13] C. Hellberg, D. Greene, T. Boyes, "Broadband Network Architectures", Prentice-Hall, 2007

[14] L. Mamakos, J. Evarts, D. Carrel, D. Simone & R. Wheeler, "A Method for Transmitting PPP Over Ethernet (PPPoE)", RFC2516, IETF, Feb. 1999

[15] J. Lau, M. Townsley, I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC3931, IETF, March 2005

[16] S. Kent & K. Seo, "Security Architecture for the Internet Protocol", RFC4301, IETF, Dec. 2005

[17] S. Kent, "IP Encapsulating Security Payload (ESP)", RFC4303, IETF, Dec. 2005

[18] S. Kent, "IP Authentication Header", RFC4302, IETF, Dec. 2005

[19] C. Kaufman, P. Hoffman, Y. Nir & P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5886, Sept. 2010

[20] B. Carpenter, "Connection of IPv6 Domains via IPv4 Clouds", RFC3056, IETF, Feb. 2001

[21] E. Nordmark & R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Router", RFC 4213, Oct. 2005

[22] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC4380, IETF, Feb. 2006

[23] L. Yang, R. Dantu, T. Anderson & R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, IETF, April 2004

[24] A. Baliga, X. Chen, B. Coskun, G. de los Reyes, S. Lee, S. Mathur, J. E. Van der Merwe, "VPMN – Virtual private mobile network towards mobility-as-a-service", Proceeding MCS, 2011

[25] J. Kempf, B. Johansson, S. Pettersson, H. Luning, T. Nilsson, "Moving the mobile evolved packet core to the cloud", WiMob 2012.

[26] http://datatracker.ietf.org/wg/dmm/