Towards Adoption of Software Defined Wireless Backhaul Networks

Osianoh Glenn Aliu[1,2], Senka Hadzic[1], Christian Niephaus[2], and Mathias Kretschmer[1,2](✉)

[1] Network Research Department, Fraunhofer FOKUS, Sankt Augustin, Germany
mathias@defutech.de
[2] DeFuTech, Hennef, Germany

Abstract. The flexibility of future wireless network architectures is aimed at allowing more innovation, reducing complexity and improving service offerings. Software Defined Networking (SDN) has been identified as an enabler for this adoption. From an implementation perspective, we provide a description of use cases and a framework overview for its implementation. This framework and the future work identified serve as pointers for further research projects in Software Defined Wireless Network (SDWN).

Keywords: Backhaul · Software defined · Flexibility · Bootstrap · Auto-configuration

1 Introduction

Immersed in various marketing and research activities on breakthrough data download speeds as the basis of the Future Internet and 5th Generation (5G) wireless networks, the real game changers will be technologies that focus on being enablers to ensure future networks are self organised, flexible and programmable. Vertical system architectures where a single manufacturer is responsible for developing the hardware, software and applications will gradually be phased out by a separation of the application layer, control layer and the data layer. One of the main drivers for this new phase in communication networks is SDN.

Network operators are eager to embrace new services and revenue opportunities, but not necessarily new technologies that require overhauling their physical network. It is thus pertinent to give operators the freedom and flexibility to innovate and create new service offerings by providing them with a more flexible network architecture. This is the main driver for the evolution of today's communication networks towards being software based. With the introduction of OpenFlow in [1], and collaboration between all major stakeholders at the Open Networking Foundation (ONF), SDN has witnessed wide scale adoption.

This paper highlights three use cases and a corresponding architectural framework that can be adopted for implementing SDWN. The final section further highlights three areas which must be considered in future implementation projects.

B. Mandler et al. (Eds.): IoT 360° 2015, Part II, LNICST 170, pp. 521–529, 2016.
DOI: 10.1007/978-3-319-47075-7_56

2 Use Cases and Applications

The possible benefits that the introduction of programmable, flexible software defined networking will have across all types of wireless networks is varied. The authors in [2] present a comprehensive survey on SDWN and virtualization. We discuss only three use cases and applications in which the introduction of SDWN will lead developers to rethink the design of future wireless network architectures.

2.1 Flexible Wireless Backhaul

One of the candidate technologies for meeting 5G cellular network performance targets is improving interference coordination schemes in very large scale Coordinated Multipoint (CoMP) transmission. Combining signals from multiple antennas requires creating a single logical cell with multiple low cost radio nodes (forwarding devices). Furthermore, the mass deployment of small cells and use of millimeter wave (with its inherent high speed but short distance) results in the necessity for flexible wireless backhaul. Implementing flexible wireless backhaul via conventional traffic engineering schemes will not suffice due to current protocol specific implementations. However, SDWN can introduce a separation of data plane from the control plane allowing compatibility with multiple traffic forwarding protocols in the control plane with MAC adaptors for simplified forwarding on the data plane.

2.2 Flexible Traffic Steering via Heterogeneous Radios

SDWN enables the possibility for smarter load balancing and flexible wireless backhauls with QoS based routing. This is achieved due to the support for media independent handover and protocol independent forwarding. Using information of location specific capacity requirements, flexible traffic steering schemes can be implemented to achieve various operator objectives. Having the knowledge of the network state, the controller can steer the traffic and thus optimize bandwidth utilization taking radio resources into account. Flow optimization within the backhaul can also be implemented with the flexibility SDWN provides. It also enables differentiated processing of flows based on the flow information (type of service, src port, dst port, etc.) for different service types. The network PHY and MAC can thus adapt its resources based on the class of service and QoS requirements of flows. This results in more flexibility and options to meet users QoS requirements by provisioning. This application is key as validity of Selected IP Traffic Offload (SIPTO) in cellular networks becomes established.

2.3 Integrating Heterogeneous Wireless Networks

SDWN defines open interfaces to manage and configure heterogeneous access networks, through coordinated efforts between ONF, OpenDaylight and the IEEE 802 OmniRAN groups. Furthermore, by abstracting multiple heterogeneous access technologies into a single access infrastructure, it is irrelevant which actual RAN technology the end user is connected to as this remains transparent. SDWN based architectures enable infrastructure sharing among multiple service providers. The end user is also able to utilize multiple radio interfaces for mobile traffic offload, where data originally targeted for cellular networks are offloaded to Wi-Fi or other complementary technologies but abstracted as a single network and technology to the user. This use case when applied to roaming scenarios between WiFi hotspots and eNodeB drastically ensures service continuity for critical applications envisaged in next generation wireless networks.

In spite of the aforementioned use cases, it must be understood that SDWN is not the magic wand that will provide solutions to all challenges in wireless communications. SDWN should be viewed as a tool that empowers us with the much needed flexibility as well as removing constraints in todays network architectures in order to innovate new solutions for existing problems and future challenges in wireless networks. Furthermore, we have encountered multiple implementation challenges which include:

1. Meeting strict QoS requirements to support provisioning of triple-play services over multi-hop, sometimes heterogeneous links
2. design tradeoff between architecture flexibility and optimal performance
3. support for heterogeneous networks with a set of technology agnostic, vendor independent primitives which are required for managing the forwarding devices

3 Unified Software Defined Wireless Networking Framework

During the course of the FP7 CARMEN project on carrier-grade wireless mesh networks', we began to investigate the programmability of simple, but heterogeneous wireless nodes and studied decentralized and centralized approaches [3]. We concluded that only a centralized controller would be able to make coherent spectrum allocation and traffic forwarding decisions in an environment potentially as volatile as the unlicensed U-NII band. Next, a technology independent messaging and addressing service was required. Extending the concept behind IEEE 8021.21 was found to be a viable solution, while the data plane required a means to enforce centrally computed data path across a multi-hop network. Multiprotocol Label Switching (MPLS) was considered a suitable and technology independent solution, among others. During the course of the SolarMesh projects [4] we developed this concept further with a strong focus on energy-efficient wireless back-hauling and arrived at a generic architecture with support for programming heterogeneous wireless interfaces. These architectures serve as the basis for what is explained in the following sections.

Figure 1 shows our proposed SDWN logical framework directly inspired from the framework used in SDN. A description of the key modules is provided as follows:

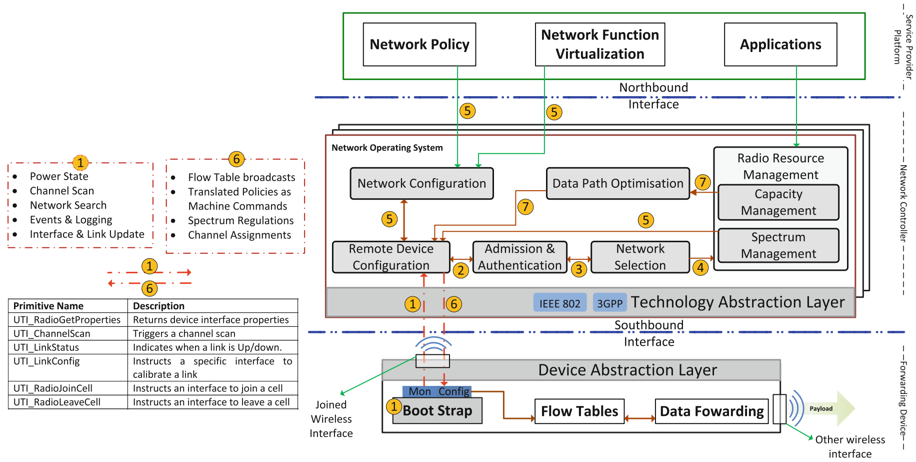

Fig. 1. SDWN logical framework

3.1 Control Plane

In wireless networks, the control plane consists of more than just a single physical controller or network entity. This is the core of the network architecture that serves as a platform for the network operating system. It orchestrates the traffic forwarding and signalling behavior of the network via its interface to the lower layers. Provisioning of network services, defining forwarding rules, traffic routes and radio resource management is enforced on the network at this layer. The control plane is responsible for enforcing polices defined in the service plane. As regards topology of the underlying infrastructure, this plane also incorporates functionalities for load balancing and Network Function Virtualization (NFV). In wireless networks, topology management is key due to new nodes joining and leaving the network on varying time scales. In addition, mobility management is required especially for infrastructure-less wireless networks.

Depending on the type of technology used in the wireless network, the control plane would consist of different network elements. For example in cellular networks its main network elements will be located in the core network, in IEEE 802 wireless[1] networks as a master node or controller managing a set of wireless access points. The framework shown in Fig. 1 is a logical representation. In practice, it is expected to utilize physically distributed instances of controllers managing different slices of the network while synchronization of all functionalities remains at a centralized point in this plane. The key modules which are technology agnostic which should be implemented on this plane include network selection, network configuration for managing forwarding devices on the data plane, and traffic routing for path computation and enforcing routing policies.

[1] WiFi, WLAN, WiMAX, etc.

3.2 Technology Abstraction Layer

An important module of the control plane is the technology abstraction layer as shown in Fig. 1. Abstraction helps reduce complexity in network design, implementation and supports interoperability. We strongly believe future wireless networks will be flexible to deliver services whereby the underlying technology (e.g. LTE, WiFi, etc.) remains transparent to the end user. This ensures support for operation over heterogeneous technologies and is achieved by having a common set of message primitives in which the control layer can seamlessly configure and manage the underlying network. Since multiple protocols can be used to implement this, an open southbound interface is defined between this layer and forwarding devices in the data plane. More information about the technology abstraction layer and device abstraction layer simply a generic abstract interface in [5] and further developed as a single universal technology interface in [6].

3.3 Southbound Interface

To ensure interoperability between vendors as well as to support heterogeneous networks, a standardized interface is specified to describe communication between devices on the data plane and network elements in the control plane. This communication is done over a secure channel and the main objective is to manage the forwarding devices or nodes, including device configuration. Various protocols can be used to achieve this, such as Forwarding and Control Element Separation (ForCES), Network Configuration Protocol (NETCONF), SNMP4SDN (an extension of SNMP), Interface to Routing System (I2RS) and Path Computation Element Communication Protocol (PCEP). Protocols used for secure control-data plane communications vary based on the type of network and use case being considered. OpenFlow is currently the most widely adopted but as it is just one protocol on the southbound interface, it is important to ensure the southbound interface remains "Open" for other protocols (existing and new) in a non-proprietary way.

3.4 Device Abstraction Layer

This layer is responsible for providing information on the type of device and capabilities of its forwarding interfaces to the control layer. A Universal Technology Interface (UTI) can be implemented both in the control and data planes as a technology abstraction layer and/or device abstraction layer, respectively as shown in Fig. 1. Depending on the type of technology and use case considered, implementation of this abstraction layer can be done in the physical controller or in the forwarding devices. One of the major proponents for its implementation in the controller is to ensure compatibility with generic forwarding devices for easier adoption. Similarly, implementation in the forwarding devices would help equipment vendors make device specific translations based on device agnostic, generic message primitives in order to improve device performance.

3.5 Data Plane

The final layer responsible for data forwarding and connecting to the end user device is the data plane. SDN concept advocates for devices that are simplified, low cost, minimal processing but specialized for forwarding packets. In essence, the devices here should primarily receive forwarding rules from the control plane and take actions based on a set of preconfigured traffic routes stored in its flow tables. Statistics of the network are periodically sent back to the control plane for optimising forwarding rules.

The peculiarity of the data plane in wireless networks requires configuration of the wireless links between forwarding devices, as well as the wireless link to their controller. It is thus pertinent to include a link configuration module in forwarding devices specifically for link monitoring and configuration. This includes simple localized functions, particularly for modules that cannot be performed efficiently from a physically centralized point. In essence, a boot strap phase will be used for initial setup connection and configuration of the device. An example is channel scan for outdoor Wireless Local Area Network (WLAN) to determine interference free channels, determining active links to neighbouring nodes, channel state information, etc.

Consider an example when a wireless node with plug and play functionality tries to connect to an already live network. The messaging flows indicated in Fig. 1 are described as follows:

1. At Bootstrap phase, the forwarding device does a self-discovery process to identify its interfaces and capabilities. A power-spectrum scan is initiated to identify free channels on its interfaces. This is done along with a network search. The scan results are encoded in a messaging format and sent out to the controller or master.
2. When the controller receives this encoded messages, it executes a security check for authentication and authorization.
3. Network selection module is responsible for selecting the type of physical network (assuming a heterogeneous network) and to which network slices this new node will be associated to.
4. Radio Resource Management (RRM) will aid in spectrum allocation based on a global view of the network, neighboring nodes and determine channel to be used on other interfaces (excluding the joined interface). This module is also responsible for DFS in networks operating on the UN-II bands, TVWS external database or licensing policies in event of operation on a licensed band as in mobile cellular networks.
5. Optimum allocations are signalled to the Remote Device Configuration (RDC) module. Simultaneously, network policies and virtualisation instances from the service provider platform aid the network configuration module to signal the RDC.
6. The RDC which translates the information via the Technology abstraction layer as simple machine language commands/instruction set to the forwarding device. Examples of 6 message primitives and their description are

summarized in Fig. 1. We have implemented these primitives and used to control the wireless nodes (see [5,6] for more details).

7. Based on statistical analysis on the types of applications and QoS requirements, the capacity management updates the data path optimisation. Updated flow tables and channel assignments for all interfaces are thus signalled to the RDC and sent as multicast.

There exists a fundamental tradeoff between achieving flexibility in the network and optimum performance, which leads to the consideration of different architectures. The key design principle however is choosing the right level of abstraction in separation of the control plane form the dataplane. The choice of these primitives highlighted in Fig. 1 was in no way trivial but based on ongoing in house development and testing which started in 2008 [3], feedback from service providers and performance benchmarking. A subset of the modules, required messaging primitives and the flow sequence have been made public in [5,7,8] by the same authors. Nevertheless, a full listing of all the primitives required, their descriptions and effect when generated will be provided in the SDN4wireless project [9]. Details of this framework for third party validation will also be made available in the project [9].

4 Future Research Directions

We present existing challenges in the form of questions that need to be addressed in order for full adoption of SDWN paradigms in the wireless domain.

Definition of Interfaces. Wireless heterogeneous networks deal with more compatibility issues from multiple technology specific protocols as compared to wired networks. The choice of a common descriptive language and command primitives set for all 'event-action'. By further defining specifications for open generic interfaces, what specialized networks functionalities may be lost? Do such heterogenous networks with common set of primitives have higher performance than existing proprietary implementations? To address these questions, more activity is expected to focus on interface description and published set of command primitives for network interface cards and to ensure protocols on these interfaces remain open and not closed to members of consortiums. There are also concerns on the level of information in the APIs provided by hardware manufacturers for third party developers.

Software Defined MACs. In order to achieve a general purpose but context aware MAC protocol, programmability of the network interface cards is required. Motivated by the attempt to program abstractions for wireless terminals, some research directions suggest MAC be made context-dependent, yielding simple programming models yet providing enough flexibility to support most customization needs. As wireless card manufacturers gradually make available their wireless MAC processor architecture to developers, more research activity and new innovations are expected on software defined MAC.

What Is the Role of Data Analytics? With the huge benefits SON had on mobile networks [10], future wireless networks in general are envisaged to incorporate more intelligence to ensure programmability from a logical central point, utilize real-time data analytics as well as incorporating Artificial Intelligence (AI) schemes. Such schemes will enable smart cities utilizing infrastructure from wireless networks. However, a lot of data will be required and statistics gathered from the forwarding layer in order to be analyzed at the service layer. This feedback enables the network to constantly self-optimize its performance, also creating personalized services for users. Further work should focus on stability of AI schemes, data analytic tools to serve as input for rule creation as well as legal issues surrounding user profiling and data gathering.

Our future work will include evaluation results which provides a comparison of our software defined wireless backhaul network with existing wireless backhaul networks. This will follow a 3 step approach listing functional module description to hardware implementation and quantitative performance evaluation.

5 Conclusion

In this paper, we have presented a logical framework for software defined wireless networks. The main impact is addressing the general question of how centralized decision making modules have to be for wireless networks. We realized that certain controller functions have to remain at the forwarding devices, i.e. initial channel scan, link status neighbor sensing. This is especially important during the bootstrap phase. Further description of a technology abstraction layer and device abstraction layer were described with references for more detailed technical description. This technology abstraction is pertinent for future heterogeneous networks, including satellite networks for ubiquitous backhaul connectivity in remote and rural areas. To support technology agnostic data forwarding devices, multilink CPE prototypes are also being investigated in [11].

We believe that the open issues, solutions and framework that have been identified and summarized in this paper, will trigger new ideas and further act as pointers to developing solutions. Software defined wireless networks will certainly create the required enablers for the next generation of wireless networks.

Acknowledgment. This work has been supported by the BATS research project which is funded by the European Union Seventh Framework Program under contract n317533. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the BATS project or the European Commission.

References

1. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)

2. Liang, C., Yu, F.: Wireless network virtualization: a survey, some research issues and challenges. IEEE Commun. Surv. Tutorials **17**(1), 358–380 (2015). Firstquarter
3. Banchs, A., Bayer, N., Chieng, D., de la Oliva, A., Gloss, B., Kretschme, M., Murphy, S., Natkaniec, M., Zdarsky, F.: CARMEN: delivering carrier grade services over wireless mesh networks. In: IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2008, pp. 1–6, September 2008
4. Horstmann, T., Kretschmer, M., Modeker, J., Niephaus, C., Sauer, S.: Development framework for prototyping heterogeneous multi-radio wireless networks. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pp. 1–5, July 2011
5. Kretschmer, M., Batroff, P., Ghinea, G.: Topology forming and optimization framework for heterogeneous wireless back-haul networks supporting unidirectional technologies. J. Netw. Comput. Appl. **36**(2), 698–710 (2013)
6. Niephaus, C., Aliu, O., Hadzic, S., Kretschmer, M., Ghinea, G.: WiBACK: a back-haul network architecture for 5G networks, In: IET International Conference on Frontiers of Communications, Networks and Applications (IET ICFCNA) (2014)
7. Niephaus, C., Kretschmer, M., Jonas, K.: QoS-aware wireless back-haul network for rural areas in practice. In: 2012 IEEE Globecom Workshops (GC Wkshps), pp. 24–29, December 2012
8. Niephaus, C., Aliu, O.G., Kretschmer, M., Hadzic, S., Ghinea, G.: Wireless back-haul: a software defined network enabled wireless back-haul network architecture for future 5G networks. In: IET Networks, September 2015. http://digital-library.theiet.org/content/journals/10.1049/iet-net.2015.0009
9. SDN4Wireless, Project deliverables, 2015–2017. www.sdn4wireless.org
10. Aliu, O., Imran, A., Imran, M., Evans, B.: A survey of self organisation in future cellular networks. IEEE Commun. Surv. Tutorials **15**(1), 336–361 (2013)
11. BATS, Project deliverables 2012–2015. http://www.batsproject.eu/