Hierarchical SDN Orchestration of Wireless and Optical Networks with E2E Provisioning and Recovery for Future 5G Networks

Ricard Vilalta[1], Arturo Mayoral[1], Jorge Baranda[1], Jose Nuñez[1], Ramon Casellas[1], Ricardo Martínez[1], Josep Mangues-Bafalluy[1], Raul Muñoz[1]
[1]*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Castelldefels (Barcelona), Spain.*
ricard.vilalta@cttc.es

Abstract: This paper proposes the hierarchical SDN orchestration of heterogeneous wireless/optical networks. End-to-End connectivity services are provisioned through different network segments by means of a Transport API. The hierarchical approach allows scalability, modularity and more security.

OCIS codes: (060.4256) Networks, network optimization; (060.4253) Networks, circuit-switched.

1. Introduction

Future 5G networks will have to be highly flexible to serve multiple services with stringent requirements such as ultra-low latency, ultra-high bandwidth, but also in terms of deployment flexibility (e.g., multi-tenancy or network density) in order to deliver End-to-End (E2E) services. These requirements can be met by efficiently integrating heterogeneous wireless and optical network segments (radio access/backhaul, aggregation, and core) and massive computing and storage services, delivered by means of Cloud/Fog computing [1].

On the one hand, a wireless SDN architecture has been proposed in [2], where a wireless SDN controller is responsible for the network control of the wireless nodes and links (e.g., mmWave, LTE, WiFi, 5G). Currently, ONF is in the process of standardizing OpenFlow (OF) extensions for wireless. On the other hand, a hierarchical SDN orchestrator has been previously introduced in order to provide an efficient solution to the network orchestration of different control plane technologies. A parent/child SDN orchestrator architecture has been previously validated for E2E multi-layer (packet and optical) and multi-domain provisioning across heterogeneous control domains (SDN/OF and GMPLS/AS-PCE) employing dynamic domain abstraction based on virtual node aggregation. The hierarchical SDN orchestrator is based on the IETF Applications-Based Network Operations (ABNO) architecture.

It is in this context where hierarchical SDN Orchestration is proposed in this paper as a feasible solution to handle the heterogeneity of wireless and optical network domains, technologies, and vendors. It focuses on network control and abstraction through several control domains, whilst using standard protocols and modules. The need of hierarchical SDN orchestration is justified not only to enable a multi-(technology, domain, and vendor) environment: but also for scalability, modularity, and security purposes.

In order to offer E2E transport service provisioning and orchestration across multiple hierarchical network segments, we propose to use a simplified Transport API [3], such as the one currently being standardized at ONF [4]. We use this Transport API as the SouthBound Interface (SBI) of a parent SDN orchestrator in order to provision E2E services towards a child SDN orchestrator, which also uses this Transport API as a NorthBound Interface (NBI).

This paper presents an architecture for hierarchical SDN orchestration of wireless and optical transport domains, using a hierarchy of SDN orchestrators. In particular, it provides an experimental evaluation of E2E provisioning and E2E recovery procedures in a multi-domain network integrating EXTREME wireless and ADRENALINE optical networking testbeds.

2. Proposed hierarchical SDN orchestration architecture for wireless/optical integration

Fig.1.a shows the proposed hierarchical SDN architecture for the integration of wireless and optical transport networks. In the wireless segment, a wireless SDN controller is in charge of the programming of the wireless mesh backhaul network, which is formed with Small Cells (SCs). This wireless SDN controller tackles the specificities of the wireless medium, implementing the proper extensions to control wireless devices. In the wired transport segment, two SDN-enabled MPLS-TP aggregation networks and a core network are controlled. The latter uses an Active Stateful PCE (AS-PCE) on top of a GMPLS-controlled optical network.

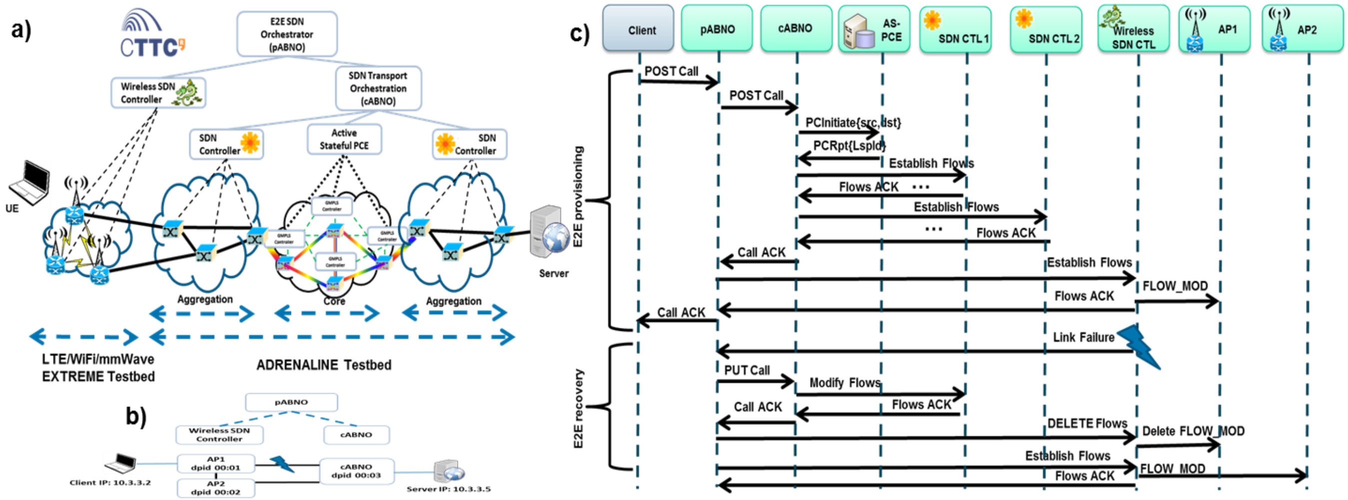


Fig.1. a) Hierarchical SDN orchestration architecture b) Network Topology view at the E2E SDN orchestrator, c) Message exchange workflow for E2E provisioning and recovery connectivity services.

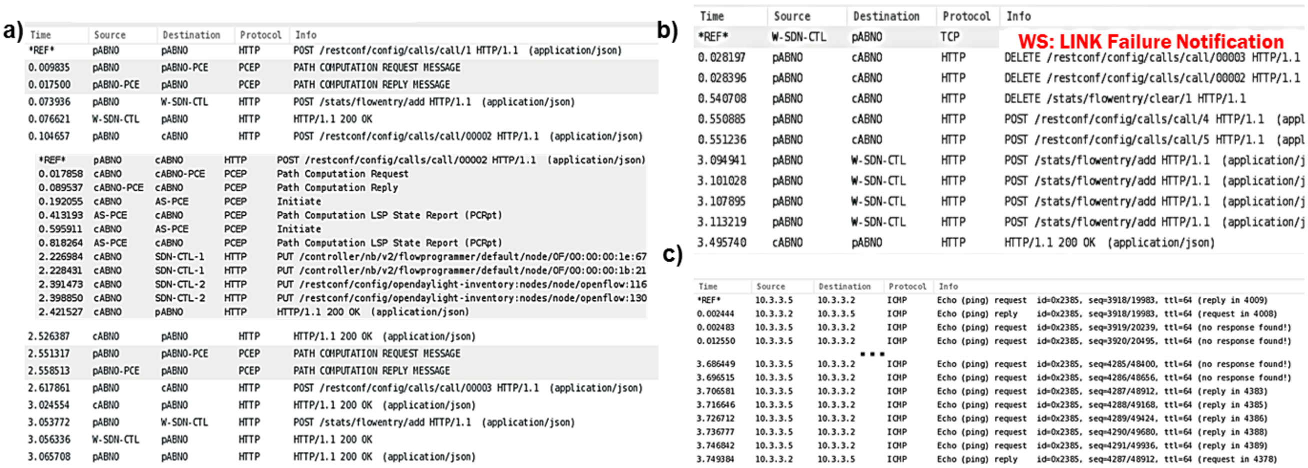
In the proposed architecture, we introduce an E2E SDN Orchestrator, responsible for the provisioning of E2E connections across different network segments. It has been implemented using the parent ABNO (pABNO) in [5]. The pABNO is able to orchestrate several network segments: an SDN-enabled wireless segment and an optical transport network segment controlled by a child ABNO (cABNO). The cABNO provides the underlying network domains abstracted as a single node. The wireless SDN controller provides the underlying wireless network elements, as well as the inter-domain links. Fig.1.b shows the topological view seen from the perspective of the E2E SDN Orchestrator.

The hierarchical ABNO architecture has been described in [5]. In addition to ABNO controller, provisioning and topology managers, VNTM and PCE, the hierarchical ABNO introduces the Abstraction Manager (AM), which is the component of a cABNO responsible for the abstraction of the underlying network resources. The proposed AM is able to provide several types of hierarchical abstraction levels, such as node or link abstraction. Node abstraction offers to the higher level of hierarchy the underlying domains/segments as an abstract node, with the input/output ports of the underlying domains/segments as ports of each abstracted node. A common Transport API, described in [3], acts as a pABNO SBI and as a cABNO NBI. This API enables the necessary hierarchical architecture in order to offer E2E connectivity, topology, and path computation services.

Fig.1.c shows the proposed message exchange between a pABNO and a wireless SDN controller/cABNO. Two different scenarios are presented: E2E provisioning and E2E recovery. In the first scenario, it can be observed that an E2E connection is requested (POST Call) to the pABNO. The pABNO computes the involved network controllers (Wireless SDN/cABNO) and requests the underlying connection. The request sent to the wireless SDN controller is directly translated into OF commands sent to the wireless SCs. Moreover, we can observe how the workflow follows inside a cABNO, which is responsible for another level of hierarchical SDN orchestration, as presented in [6]. The cABNO first requests an optical lightpath to the AS-PCE and, once the lightpath has been established, the different flows are established towards the underlying packet SDN controllers. In the second scenario, an inter-domain link failure is detected by the wireless SDN controller. The link failure is notified to the E2E SDN orchestrator, that computes a new E2E path, and, following a “break-before-make” strategy, it first deletes or modifies the old connections and then establishes the new computed recovery connections.

3. E2E provisioning and recovery results in a multi-domain multi-technology wireless/optical testbed

The EXTREME SDN wireless testbed is based on 12 programmable SCs based on PC (Intel core i7 and 32GB RAM). Each SC is equipped with up to three Compex WLEV900X 802.11ac Atheros cards (ac/a/b/g/n) and Ethernet ports, also used to set up inter-domain links. Each SC runs xDPd as software switch and the wireless SDN controller is based on Ryu. Two SCs are interconnected to different packet switches, providing two inter-domain links between the EXTREME and ADRENALINE testbeds (Fig.1.a). The ADRENALINE Testbed consists of two packet SDN/OF domains based on Ethernet transport technology and an optical transport network. Each packet domain consists of 2 OF switches deployed using COTS hardware and running OVS. Each OF border switch includes a 10 Gb/s XFP tunable transponder. The transport network consists of a GMPLS-controlled WSON centrally managed by an AS-PCE.



a)

```
Time      Source     Destination Protocol Info
*REF*     pABNO      pABNO       HTTP     POST /restconf/config/calls/call/1 HTTP/1.1  (application/json)
0.009835  pABNO      pABNO-PCE   PCEP     PATH COMPUTATION REQUEST MESSAGE
0.017500  pABNO-PCE  pABNO       PCEP     PATH COMPUTATION REPLY MESSAGE
0.073936  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/json)
0.076621  W-SDN-CTL  pABNO       HTTP     HTTP/1.1 200 OK
0.104657  pABNO      cABNO       HTTP     POST /restconf/config/calls/call/00002 HTTP/1.1  (application/json)

*REF*     pABNO      cABNO       HTTP     POST /restconf/config/calls/call/00002 HTTP/1.1  (application/json)
0.017858  cABNO      cABNO-PCE   PCEP     Path Computation Request
0.089537  cABNO-PCE  cABNO       PCEP     Path Computation Reply
0.192055  cABNO      AS-PCE      PCEP     Initiate
0.413193  AS-PCE     cABNO       PCEP     Path Computation LSP State Report (PCRpt)
0.595911  cABNO      AS-PCE      PCEP     Initiate
0.818264  AS-PCE     cABNO       PCEP     Path Computation LSP State Report (PCRpt)
2.226984  cABNO      SDN-CTL-1   HTTP     PUT /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:1e:67
2.228431  cABNO      SDN-CTL-1   HTTP     PUT /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:1b:21
2.391473  cABNO      SDN-CTL-2   HTTP     PUT /restconf/config/opendaylight-inventory:nodes/node/openflow:116
2.398850  cABNO      SDN-CTL-2   HTTP     PUT /restconf/config/opendaylight-inventory:nodes/node/openflow:130
2.421527  cABNO      pABNO       HTTP     HTTP/1.1 200 OK  (application/json)

2.526387  cABNO      pABNO       HTTP     HTTP/1.1 200 OK  (application/json)
2.551317  pABNO      pABNO-PCE   PCEP     PATH COMPUTATION REQUEST MESSAGE
2.558513  pABNO-PCE  pABNO       PCEP     PATH COMPUTATION REPLY MESSAGE
2.617861  pABNO      cABNO       HTTP     POST /restconf/config/calls/call/00003 HTTP/1.1  (application/json)
3.024554  cABNO      pABNO       HTTP     HTTP/1.1 200 OK  (application/json)
3.053772  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/json)
3.056336  W-SDN-CTL  pABNO       HTTP     HTTP/1.1 200 OK
3.065708  pABNO      pABNO       HTTP     HTTP/1.1 200 OK  (application/json)
```

b)

```
Time      Source     Destination Protocol Info
*REF*     W-SDN-CTL  pABNO       TCP      WS: LINK Failure Notification
0.028197  pABNO      cABNO       HTTP     DELETE /restconf/config/calls/call/00003 HTTP/1.1
0.028396  pABNO      cABNO       HTTP     DELETE /restconf/config/calls/call/00002 HTTP/1.1
0.540708  pABNO      cABNO       HTTP     DELETE /stats/flowentry/clear/1 HTTP/1.1
0.550885  pABNO      cABNO       HTTP     POST /restconf/config/calls/call/4 HTTP/1.1  (appl
0.551236  pABNO      cABNO       HTTP     POST /restconf/config/calls/call/5 HTTP/1.1  (appl
3.094941  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/j
3.101028  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/j
3.107895  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/j
3.113219  pABNO      W-SDN-CTL   HTTP     POST /stats/flowentry/add HTTP/1.1  (application/j
3.495740  cABNO      pABNO       HTTP     HTTP/1.1 200 OK  (application/json)
```

c)

```
Time      Source    Destination Protocol Info
*REF*     10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=3918/19983, ttl=64 (reply in 4009)
0.002444  10.3.3.2  10.3.3.5    ICMP     Echo (ping) reply    id=0x2385, seq=3918/19983, ttl=64 (request in 4008)
0.002483  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=3919/20239, ttl=64 (no response found!)
0.012550  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=3920/20495, ttl=64 (no response found!)
                          ...
3.686449  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4285/48400, ttl=64 (no response found!)
3.696515  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4286/48656, ttl=64 (no response found!)
3.706581  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4287/48912, ttl=64 (reply in 4383)
3.716646  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4288/49168, ttl=64 (reply in 4385)
3.726712  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4289/49424, ttl=64 (reply in 4386)
3.736777  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4290/49680, ttl=64 (reply in 4388)
3.746842  10.3.3.5  10.3.3.2    ICMP     Echo (ping) request  id=0x2385, seq=4291/49936, ttl=64 (reply in 4389)
3.749384  10.3.3.2  10.3.3.5    ICMP     Echo (ping) reply    id=0x2385, seq=4287/48912, ttl=64 (request in 4378)
```

Fig.2. a) E2E provisioning wireshark captures at pABNO and cABNO, b) E2E recovery wireshark captures at the pABNO, c) Data plane E2E recovery between UE and Server.

The implementation of the ABNO uses Python for most of the components and C++ for PCE component. Several internal REST interfaces are offered between the different components. In this paper we have used the Transport API to interact between pABNO and cABNO. The Transport API information model is described in YANG, and it uses RESTconf as transport protocol. RESTconf allows using the Transport API as an HTTP REST API. As a REST API does not allow notifications, websockets have been introduced (as described in RESTconf) in order to send notifications of nodes and links status.

Fig.2.a shows the wireshark captures of the exchanged messages at both the pABNO and cABNO. The bidirectional E2E service call request is received at the pABNO. The pABNO PCE is responsible for computing an E2E path. The pABNO VNTM decomposes the computed E2E path in order to request a call service to the cABNO. The cABNO is responsible for SDN orchestration towards the underlying network domains (SDN-CTL-1, AS-PCE, and SDN-CTL-2). After the cABNO provisions the requested call service, the pABNO requests the necessary flows to the wireless SDN controller. The setup delay for an E2E service call is around 3.06s.

The E2E recovery message exchange is shown in Fig.2.b. The wireless SDN controller receives the OF port down message in an inter-domain link and notifies the pABNO of the link failure through an open websocket. The pABNO computes the E2E recovery path and it first removes the previously provided connections. Then, it establishes the new connections between the wireless nodes.

Fig.2.c. shows the data plane connectivity between a User Equipment (UE) connected through a wireless SC and a server running at the aggregation network. ICMP ping messages (with a message interval of 10ms) are exchanged between both hosts. We emulate a link failure by disconnecting one of the inter-domain ports. When the E2E recovery is applied, the ICMP request messages are recovered and replied. The resulting E2E recovery time from the data plane perspective is around 3.7s.

4. Conclusions

We have presented E2E provisioning and recovery use cases in an integrated optical/wireless testbed by means of hierarchical SDN orchestration. Experimental results suggest that certain levels of hierarchy can cope with the upcoming network heterogeneity from a multi-(technology, domain, and vendor) perspective.

5. Acknowledgments

The research leading to these results has received funding from EC H2020 XHAUL project (H2020-671598) and Spanish MINECO projects FARO (TEC2012-38119) and 5GNORM (TEC2014-60491-R).

6. References

[1] R. Muñoz, et al, Towards an SDN/NFV-based multi-tenant network and cloud testbed for end-to-end 5G services, WWRF 34, 2015.
[2] C. J. Bernardos, et al., An architecture for software defined wireless networking, IEEE Wireless Communications 21.3, 2014.
[3] R. Vilalta et al., The Need for a Control Orchestration Protocol in Research Projects on Optical Networking, EuCnC, 2015.
[4] OIF-ONF white paper, Global Transport SDN Prototype Demonstration, 2014.
[5] R. Vilalta, et al., Hierarchical SDN Orchestration for Multi-technology Multi-domain Networks with Hierarchical ABNO, ECOC 2015.
[6] R. Muñoz, et al., Transport Network Orchestration for end-to-end Multi-layer Provisioning Across heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains, IEEE/OSA JLT 2015.