Control, Management and Orchestration of Optical Networks: An Introduction, Challenges and Current Trends

Ramon Casellas(1), Ricardo Martínez(1), Ricard Vilalta(1) and Raül Muñoz(1)

(1) Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), ramon.casellas@cttc.es

Abstract *This tutorial is an introduction to control and management; focusing on main drivers, key benefits and functional/protocol architectures. It covers multi-domain and multi-layer networks and includes complex use cases and current trends such as joint IT/network orchestration and slicing*.

Introduction and main requirements

A telecommunications network is composed of network elements (NE), interconnected by transmission links. Such elements, which may be either circuit or packet switching, switch / forward data based on a set of implicit or explicit parameters: a *programmable* generic switch matches incoming frames looking up a set of rules - across multiple headers and layers - and forwards them performing actions such as transformations, encapsulations, or replications, and an OXC cross-connects e.g. a frequency slot from an input port to an output port. In this context, to provision a network service (e.g., a data connection), a path needs to be computed, resources pre-assigned and subsequently reserved, forwarding rules defined and NEs configured. For this, NEs offer elements and interfaces that provide operation, management, monitoring and configuration services.

Automation as Overall Main Requirement

A straightforward requirement is automating the provisioning of such services cost-effectively, allowing autonomic network operation and empowering users to efficiently control allocated resources, minimizing manual intervention. Such process needs to be done across the whole network - with increasing traffic dynamicity requiring frequent and complex re-arrangements within multiple technological layers and in networks spanning multiple segments – e.g. by means of a management system[1] (NMS). A separated management network enables centralized provisioning, seen as a sequence of operations for configuration and state definition. Such *Permanent Connections* reflect longer timescales and lower dynamicity. Protocols for such purpose (e.g., SNMP) are low level, lack desired flexibility, expressiveness and do not support advanced functions such as remote procedure calls, so a logical operation can turn into a sequence of interactions keeping state until the operation is complete, and if error, needing to roll the device back into a consistent state.

Introducing the Control Plane

There is debate whether a sufficiently developed management plane (MP), with augmented interfaces can indeed provide such automation meeting all the requirements. In short, both planes co-exist with a given functional split: the MP conceptually focuses on FCAPS[1], including the configuration of the CP itself, delegating the actual provisioning to it in a “top-down”, *separation-of-concerns* approach. In transport networks, the CP was introduced as a means to ease operation (e.g. automatic discovery), off-loading the MP and simplifying the service provisioning process while, at the same time, leveraging the benefits of decentralized routing and control, such as path protection in arbitrary meshed networks and adaptive traffic engineering, having standard interfaces that could enable interoperability. The CP is thus the system and functions that covers the dynamic and on-demand provisioning of network services between endpoints, configuring associated switching and forwarding state, and including the functions not originally part of a NMS, where inventory and topology are manually managed.

The design of a CP involves a set of entities that inter-communicate, defined within functional and protocol architecture(s). It needs to address ever-growing requirements related QoS/QoT, remaining valid in a multi-domain and multi-layer environment and, for optical networks, ideally accounting for constraints such as the effect of physical impairments, quality of transmission, or wavelength continuity constraints. Emerging requirements include e.g., extending the scope to include geographically disperse datacenters; managing the allocation of heterogeneous resources from computing, networking and storage domains and supporting 5G/IoT networks and associated business models and services, such as slicing or network virtualization.

Distributed and Centralized Control Models

Distributed models have their roots in the design of IP dynamic routing and later on the IP/MPLS control plane, assuming administrative regions loosely tied with changing interconnections as traffic fluctuates and failures occur, and exemplified by the ASON/GMPLS architecture[2].

On the other hand, in centralized models a controller interacts with CP agents located in the nodes, and CP logic remains in the controller. The latter are justified by their (relative) simplicity, addressing the shortcomings of distributed control planes. In any case, a control plane must support a set of basic functions, including addressing, interface and resource management and discovery, topology and reachability management, path computation and service provisioning with recovery. Both models present their strengths and weaknesses: a central control is conceptually simpler, a single point of deployment of policies, and business logic, easier to deploy APIs, and requires less state synchronization. It may also present a bottleneck or single point of failure, with potential fault-tolerance issues. On the contrary, some functions (dynamic restoration, fast rerouting) are difficult to achieve in a centralized model, and a distributed CP is more robust and mature, although implementations usually need to conform to a wider set of protocols. It may also operate independently of the NMS, although it is not the default mode of operation. In short, hybrid approaches are to be expected, integrating both depending on actual deployment requirements.

Software Defined Networks

SDN[3] is simplistically defined as a centralized control model architecture and protocols based on a clear CP and DP separation, enabling an application layer. OpenFlow was thus a particular case of standard interface and protocol leveraging programmability and exploiting the fact that most modern NE can be abstracted identifying a common set of functions, e.g. the concepts of flows, match and action tables.

A better characterization of SDN involves identifying opportunities for a better integration with OSS / BSS, addressing dynamic computing and storage needs, and easing implementation of network-wide policies, and new business models. This, within a major trend of *softwarization*, for use cases well beyond re-implementing distributed control plane logic. It relies on a systematic approach to resource management in heterogeneous contexts (cfr. *Orchestration*), with *i)* interface definition around standardized data models, *ii)* the use of unified protocol frameworks overcoming known limitations supporting network-wide transactions & rollback and *iii)* the availability of open source software of key aspects of system development. Last, this decoupling of hardware and software is allowing vendor-neutral disaggregated deployments, exploiting the capabilities of hardware to be programmed, enabling an application ecosystem.

SDN core principles can thus be broadly applied and specific control plane deployments become part of a wider SDN-*based* service and resource orchestration system[4].

Multi-Layer Networks

ML networks involve multiple technologies (e.g., a packet switched layer and a circuit switched layer) or multiple levels within a given technology. Services are understood within a client-server model where a lower layer connection supports multiple higher layer connections, enabling grooming and multiplexing. A ML CP implies being able to provision services across multiple layers. A basic model can be defined where each layer has its own CP instance, with little to no interaction (e.g. *overlay*), explained by current market segmentation (vendors), but lacks a joint control of multiple layers enabling efficient resource usage (e.g., having topology visibility of all the layers to attain optimal path computation). Other interconnection models with full topology visibility (e.g., the *peer* model) are significantly more complex. The trend is to roll out hybrid

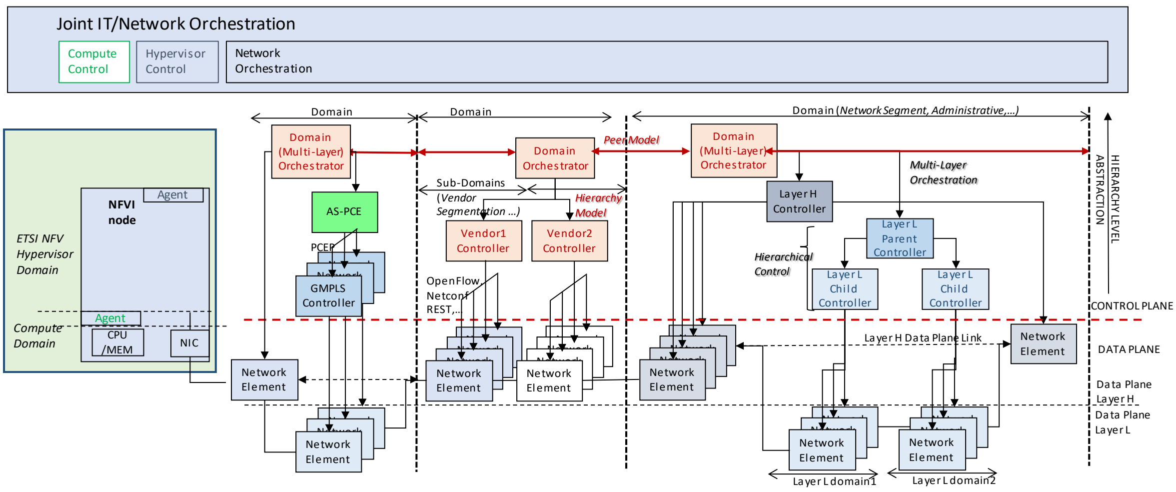

Fig. 1: Overarching Control, Management and Orchestration

models in which each layer operates independently to a large extent, yet there is some abstracted information exchanged and inter-layer coordination ensuring efficient resource usage.

Multi-Domain Networks

Transport networks are increasingly segmented in domains, e.g., to enhance scalability or due to confidentiality reasons. *Domain* admits multiple definitions, e.g. management boundaries, vendor or technology islands, or path computational responsibility. A CP challenge for multi-domain networks is the inherent limited topology visibility outside a given domain and interoperability issues for cross-domain signaling. Exchange of topological information between domains is limited to the dissemination of reachability yielding sub-optimal choices and domain local optimality does not imply end-to-end optimality. Amongst different interconnection models, a common trade-off is to rely on a hierarchical arrangement of controllers, along with some degree of topology abstraction and aggregation, minimizing interoperability and ownership issues.

Orchestration

Orchestration is often used in different contexts and may imply i) the selection of resources to satisfy service demands in an optimal way, where the available resources, the service demands and the optimization criteria are all subject to change[3], ii) the coordination of the resources needed to set up cloud-based services and applications, uses a variety of virtualization software and industry standard hardware[5] or, iii) the coherent coordination of heterogeneous systems, allocating diverse resources and composing functions to offer end-user services, automating processes and using or invoking the programming interfaces of subordinate or external systems, platforms and infrastructures.

Network Orchestration refers to heterogeneous network domains (in terms of control and/or data plane technologies), admitting diverse controller arrangements. For example, in a hierarchical setting a centralized controller of controllers or *orchestrator* automates connectivity provisioning at a higher, abstracted level, covering inter-domain aspects. Specific per-domain (child) controllers map the abstracted control plane functions into the underlying CP technology. *Joint IT/Network Orchestration*[6] refers to services requiring resources such as computing and networking domains, exemplified by NFV and the use of virtualized servers requiring connectivity, where the service provisioning process no longer stops at the physical node and needs to interact with whatever mechanism virtualization hypervisor offers. Macroscopically, orchestration solutions need to rely on abstraction (selection of an entity relevant characteristics, based on targeted functionality and scalability) and a hybrid combination of centralized and distributed entities while relying on a uniform approach to resource management (see Fig.1).

Ongoing Trends and Conclusions

The raise of SDN is bound to a significant increase of unified and systematic information and data modelling activities (effort across SDOs to model multiple aspects, including network topologies or describing device capabilities, attributes, operations, state and notifications). Optical networks are particularly challenging in this regard due to the lack of agreed-upon hardware models. Notwithstanding, cross-vendor initiatives (e.g., OpenROADM[8]) and the raise of disaggregated/white boxes are mitigating this. Another challenge is the lack of a common model for physical impairments and their effects that can be directly mapped to CP protocols.

The aforementioned *softwarization* is behind the increase of network instrumentation, monitoring and telemetry, enabling control closed-loops and continuous optimization. Additionally, there is a steady increase on analysing the potential benefits of machine learning, artificial intelligence when applied to transport networks. Finally, there is a trend to support slicing[7], that is, allocating a physical or logical part of an infrastructure and related functions to clients (e.g. virtual operators) ultimately allowing the entire control of the slice.

To conclude, the provisioning of services involving heterogeneous resources needs to be automated, with stringent requirements in terms of QoS, latency, bandwidth, enabling automatic recovery. This will require the integration of Control, Management and Orchestration systems – for a given functional split – involving hybrid deployments combining centralized and distributed elements, applied in a heterogeneous environment across multiple technological and administrative domains.

Acknowledgements

Work is partially funded by the Spanish MINECO DESTELLO project (TEC2015-69256-R) and the EC through H2020-ICT2016-2 METRO-HAUL (761727).

References

[1] ITU-T, TMN Rec Series M.3000, 2001.
[2] R. Casellas, et al "A Control..." Comm Mag, v54, n8, 2016.
[3] ONF TR-522 "SDN Arch for Transport Networks", 2016.
[4] R. Casellas et al "Overarching…", JLT v33, n5, 2015.
[5] ETSI, Network Functions Virtualisation, an Intro…2012.
[6] R. Muñoz et al, "Tr. Net Orchestration", JLT v33, n8, 2015
[7] R. Vilalta et al. "Multidomain Net…", JOCN v7, n11, 2015
[8] Open ROADM Multi-Source Agreement (MSA), *online*