Real-Time Traffic Management in AT&T's SDN-Enabled Core IP/Optical Network

Simon Tse and Gagan Choudhury
AT&T, 200 S. Laurel Ave, Middletown, NJ 07748
st2196@att.com, gc2541@att.com

Abstract: In late 2017, AT&T deployed an SDN controller that manages real-time traffic routing using a highly efficient optimization engine for its core IP network. The controller is being expanded to include optical transport layer, allowing network capacity to be dynamically adjusted in an integrated multi-layer manner. This results in a highly efficient network that dynamically adjusts to meet traffic growth, traffic shifts and network failures. This paper highlights some key learnings from the implementation and potential future work.

OCIS codes: (060.4265; 060.4256)

1. Introduction

AT&T began its network transformation a few years ago with the Domain 2.0 initiative that includes Network Function Virtualization, SDN and Open Network Automation Platform (ONAP) for controlling the virtualized infrastructure with well documented economic and flexibility benefits[1, 2]. AT&T's Common Backbone network (CBB) is a mission critical, large-scale IP/multiprotocol label switching (MPLS) network that carries all of the core IP traffic with a very high degree of reliability and performance. Instead of a complete switch-over to SDN overnight, in late 2017 we adopted a hybrid and evolutionary approach, introducing a centralized SDN controller on top of the existing distributed routing mechanism for Layer 3 traffic management. We are also expanding this controller to the Layer 0 optical transport network, initially through a vendor-provided ROADM (Reconfigurable Optical Add-Drop Multiplexer) network controller to further take advantage of an integrated multi-layer control and optimization. This strategy not only allows seamless migration from the existing network, it also permits incorporation of new open-source interoperable commercial-off-the-shelf (COTS) equipment or whiteboxes as they become available. In this paper, we describe some key learnings from the implementation.

2. SDN Architecture

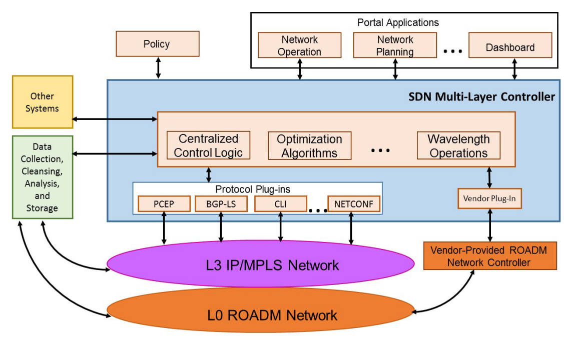

Figure 1: Schematic of SDN Network Architecture

Our implementation of an SDN Multi-Layer Controller (SDN-MLC) can be depicted schematically in Figure 1[3]. The OpenDaylight (ODL-based) SDN-MLC uses Netconf protocol for the South Bound Interface (SBI) to manage the network's data plane. In this case, the SDN-MLC controls the Layer 3 IP/MPLS router network and also the Layer 0 ROADM network initially via a vendor-provided network controller with an embedded vendor plug-in. As ROADM technology has evolved to Open ROADM with standard APIs and interoperability [4, 5, 6], the SDN-MLC can control the open ROADM network with much greater flexibility and lower cost. AT&T is currently working with other carriers and vendors on the OpenROADM MSA to define interoperable ROADM and optical plug-ins using YANG models [4] and on Transport PCE to specify Open ROADM Controller [5]. On the north-bound side via a RESTful interface, the SDN-MLC interacts with users through various applications for provisioning, monitoring, additional traffic and capacity management, etc.

Taking real-time network traffic and topology information from the control plane using the SDN controller APIs of BGP-LS/PCEP from Layer 3 and via the Layer 0 ROADM controller, SDN-MLC sends the information to our internal optimization algorithmic engine to assess, evaluate and ultimately provide recommendations via a PCEP interface for Layer 3 routing and the ROADM controller for wavelength management in the network. For example, Border Gateway Protocol-Link State (BGP-LS) gives information on link capacities/weights, and Path Computation Element Communication Protocol (PCEP) gives information on current label-switch paths (LSPs) or tunnel routing and bandwidth usage. The Algorithm uses the BGP-LS and PCEP information as well as other auxiliary information to decide on LSP changes (bandwidth or path) and/or wavelength circuit addition or deletion. While each of these changes might be simple to execute individually, the decision on the timing, priorities, and combination of recommendations can be complex such that the Algorithm must be highly intelligent, efficient and responsive.

3. Key Learnings from Scalable SDN Implementation

Need for Highly Efficient Information Processing and Execution – AT&T's CBB is a nation-wide, high capacity and dynamic network where the network condition can change in an instant, reflecting traffic fluctuations, and/or a network failure. Before SDN, we typically accounted for this through careful planning with extra capacity for LSPs or tunnels to handle traffic surges and back-up tunnels for fast re-route (FRR) in the immediate aftermath of a failure. In a hybrid implementation, we continue to maintain these practices but with added capabilities and flexibility from SDN-MLC for better resource allocation and optimization in a coordinated manner. FRR typically reacts in the order of tens of milliseconds when a network failure is detected. Afterwards, constrained shortest path first (CSPF) convergence takes place in the order of a few seconds to stabilize the network. With a distributed routing mechanism, each head-end router normally analyzes the current network topology and performs distributed local routing optimization every few minutes. Therefore, if an SDN controller is implemented in a hybrid and coordinated manner, it is important to have very high efficiency for the entire end-to-end process from network data gathering, assessment and optimization decision by centralized algorithm and finally network changes executed by the data plane in the order of a minute or so. Given the size of our core network and different classes of services, the data volume is large and consequently the optimization is complex and potentially time consuming. The execution of physical network changes from simple path adjustments to Layer 0 wavelength manipulation varies greatly. Careful process design with good policy, efficient optimization engine and reliable execution with accurate feedback are keys to a successful scalable implementation. We deployed SDN-MLC in our network with a typical cycle time of significantly sub-minute in normal network conditions to an order of a minute with many network changes.

Layer 3 Routing Stability in a Hybrid network – Layer 3 traffic routing among a specific pair of core routers and with a specific service level uses a logical path (LSP or tunnel) to carry all 5-tuple (source/destination IPs, protocol, source/destination ports) traffic flows belonging to this LSP. If needed, a specific LSP may be split into multiple ones for flexibility of routing. These logical paths are typically determined by CSPF algorithm and can change from time to time reflecting the current traffic and network conditions. A stable routing would keep the same route for the same or similar traffic flow as long as possible. Within an SDN controller, the optimization engine is sufficiently sensitive to react to any network changes but stable enough not to make changes from minor perturbations. Likewise, the distributed routing should also exhibit stable routing behavior. Here, we are further concerned about the stable routing in a hybrid environment. A poor and inconsistent design can lead to network instability with repeated routing switching when two master mechanisms take turns to adjust the routes [7]. In this recent paper, we examined what conditions would allow a stable routing pattern or how to prevent or overcome flip-flop routing. There are both theoretical and practical answers to this question. An obvious theoretical view of achieving a stable network routing is to make sure that these two mechanisms think and act alike. In this case, if both mechanisms use the same CSPF algorithm, the routing decision from either mechanism will be the same. However, the situation is not quite ideal. The distributed routing is locally optimized while centralized control is globally optimized. The traffic and network topology information available to both mechanisms are close but not always identical due to timing and occasionally incomplete information. Finally, a highly accurate optimization algorithmic scheme is desirable but it is too slow. We eventually implemented a heuristic and multi-faceted solution. We first ensure both mechanisms exhibit stable routing by themselves. Individual routers are instructed to surrender control to the centralized controller under most and normal conditions. For the SDN controller, the optimization algorithm is deployed in the most time efficient way with the ability to monitor, react and/or back-off during various scenarios.

Layer 0 Control of Optical Transport – We have discussed so far how an SDN controller is implemented to manage the core routers for Layer 3 optimized routing of LSPs (that are aggregates of 5-tuple packetized traffic flows) with existing physical network topology and capacity. While this SDN implementation is the first key step and has already resulted in improvements, one can extract much bigger benefits by extending SDN's control over the optical transport layer. That is, if an SDN controller (with the optimization engine) is enabled to create and rearrange wavelengths, it can in real-time adjust network capacity and topology to best meet the needs. For our initial multi-layer implementation, we are converting the embedded flexible Colorless/Directionless ROADM backbone network to be SDN-enabled such that our SDN-MLC can control the flexible C/D ROADMs via the vendor-provided controller for real-time wavelength addition, deletion and re-routing with constraints. This real-time wavelength management takes place in minutes greatly reducing the cycle time of the current static process.

Comprehensive and Efficient Optimization Engine – While the centralized SDN controller provides the critical command and communications with the data plane, the embedded optimization engine is the brain and real control of the entire automation. The above-mentioned learnings already demand the optimization engine to be highly efficient, comprehensive and accurate. The optimization engine needs to act as a central command unit that coordinates all intelligence and decision-making activities in an orderly and cohesive manner. For example, in normal operation, this engine needs to track what network recommendations/changes are executed, to decide how and when to repeat or back-off requests. If the SDN-MLC server fails and an automatic failover to a standby server is activated, the engine needs to monitor and proactively re-delegate the control to the operational server, as SDN-MLC typically manages a large number of routers. Incomplete or premature switch-over will render incorrect traffic information and delayed execution and that in-turn will jeopardize the network stability.

Enhanced Monitoring and Real-Time Network Analytics – Before SDN, the network is continuously monitored with alarms for trouble shooting and restoration. Network traffic data is collected periodically and sometimes on a sample basis mostly for off-line analyses. The automation nature of SDN controller with streaming data has provided much information richness and opportunities for realistic analyses to understand the network dynamics. What-if simulations with real-time data further improve the management of network maintenance and failure events.

4. Future Work

New Router and ROADM Technology Migration – The paper so far covers our initial SDN-MLC deployment with existing network architecture and technologies. As we continue our journey with the Domain 2.0 initiative, the network nodes are also evolved to a cloud architecture with whitebox or virtual routers for layer 3 and open ROADMs including other optical plug-ins for layer 0 [4-6]. Figure 2 illustrates the architecture of an open source based Open ROADM controller [5].

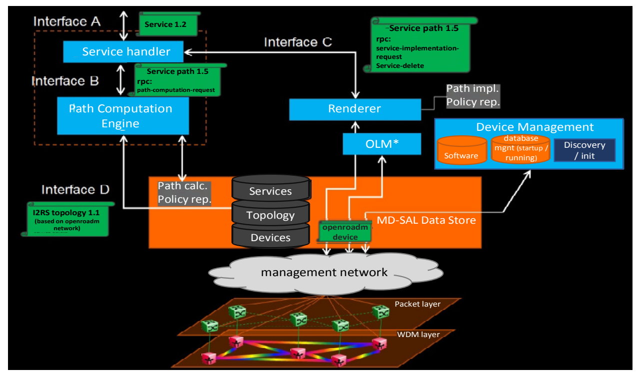

Figure 2: Schematic of Open ROADM Controller Architecture

Machine Learning for Better Network Management – The current SDN-MLC implementation with an embedded optimization engine highlights the use of real-time data to drive an automated routing or wavelength management decision. Nevertheless, it is a reactive use of the real-time information to address the network changes and optimize the resources. Machine learning techniques will further elevate the data-driven analytics for proactive planning, driving network changes, and improving network performance and reliability.

Flexible Capacity Augmentation Planning – Unlike the current static and long capacity planning cycle, the deployment of SDN capabilities fundamentally changes the capacity augmentation. While planned resources will need to go through normal business cycles, real-time wavelength or capacity turn-up/turn-down with SDN effectively shortens provisioning cycle time. At the same time, previous dedicated resources in the pre-SDN world now could become shared resources that can be used and reused continuously reacting to network failures and traffic demands. It offers opportunity to combine and simplify the current separate layer planning processes.

5. References

[1] John Donovan and Krish Prabhu, "Building the Network of the Future", CRC Press, First Edition, ISBN 1138631523, September 2017.

[2] Gagan Choudhury, Martin Birk, Bruce Cortez, Alvin Goddard, Narayan Padi, Kathy Meier-Hellstern, John Paggi, Aswatnarayan Raghuram, Kathy Tse, Simon Tse and Andrew Wallace, "Software Defined Networks to greatly improve the efficiency and flexibility of Packet IP and Optical Networks", ICNC 2017, January 26-29, 2017, Silicon Valley, CA, USA.

[3] Matin Birk, Gagan Choudhury, Bruce Cortez, Alvin Goddard, Narayan Padi, Aswatnarayan Raghuram, Kathleen Tse, Simon Tse, Andrew Wallace and Kang Xi, "Evolving to an SDN-Enabled ISP Backbone: Key Technologies and Applications", in IEEE Communications Journal, October 2016.

[4] Open ROADM MSA, *http://openroadm.org/home.html*

[5] Transport PCE, *https://wiki.opendaylight.org/view/TransportPCE:Main*

[6] Martin Birk, "AT&T's Direction Towards a Whitebox ROADM", Open Network Summit 2015, Session: SDN for Service Provider Networks: Transport SDN, June 17th 2015, Santa Clara, CA, USA.

[7] Shih-Hao Tseng, Ao Tang, Gagan Choudhury, and Simon Tse, "Routing Stability in Hybrid Software-Defined Networks", submitted to IEEE/ACM Transactions on Networking.