Chapter 16
Future SDN-Based Network Architectures

Evangelos Haleplidis
University of Patras, Greece

Spyros Denazis
University of Patras, Greece

Odysseas Koufopavlou
University of Patras, Greece

ABSTRACT

Networking has seen a burst of innovation and rapid changes with the advent of Software Defined Networking (SDN). Many people considered SDN to be something new and innovative, but actually SDN is something that has already been proposed almost a decade ago in the era of active and programmable networks, and developed even before that. Coupled with the fact that SDN is a very dynamic area with everyone trying to brand their architecture, research or product as SDN has defined a vague and broad definition of what SDN. This chapter attempts to put SDN into perspective approaching SDN with a more spherical point of view by providing the necessary background of pre-SDN technologies and how SDN came about. Followed by discussion on what SDN means today, what SDN is comprised of and a vision of how SDN will evolve in the future to provide the programmable networks that researchers and operators have longed for for many years now. This chapter closes with a few applicability use cases of the future SDN and wraps up with how SDN fits in the Future Internet Architectures.

INTRODUCTION AND HISTORICAL BACKGROUND TO SDN

Network research has gained a burst of activity and innovation for the past couple of years, with the advent of what is called, Software Defined Networking (SDN). SDN, as a term, was introduced in 2008 by Stanford University researchers (McKeown et al. 2008) as an attempt to enable researchers to operate network in a more programmable fashion in order to run their experiments such as new protocols, interfaces or algorithms on real production networks.

DOI: 10.4018/978-1-4666-8371-6.ch016

Simulations and emulations can only provide insights of whether a proof of concept may be applicable. To actually deploy new protocols or architectures on real hardware, the students would either have to convince hardware manufacturers to adopt them, but they are very reluctant to do so, as the design cycle of a new device could take a lot of time and yet generate a limited monetary incentive. Or they would have to develop the hardware themselves, using custom-based hardware as described by Lockwood et al. (2007). This custom-based hardware was soon embraced by the networking community.

However, based on research and demonstration of SDN-enabled technologies, the industry realized that by utilizing the concepts proposed by the SDN proponents, they could solve real-world problems. In environments such as data centers where it is crucial to optimize resources and thus the capability to customize the behavior of the network, till then constrained to the decision of a distributed control plane, and the capability of automation of configuration in environments was a key factor.

SDN initially begun with the precept of separating the forwarding plane from the control plane (these terms will be elaborated further in this chapter) to allow applications to program the network. The separation is achieved by abstracting the forwarding plane and providing an open interface to the control plane. Such a separation incurs many benefits to both planes as it allows research and innovation to occur independently in each plane.

However the concept of separating the control plane from the forwarding plane, or in other words, separating the signaling from the data path, has been present in the networking world for a long time documented by Feamster et al. (2013) and Mendonca et al.(2013). As discussed in Feamster et al. (2013) and later in this chapter, the main reasons for adoption was the urgent need for programmability, especially in DCs, while using open standard interfaces and utilizing existing switch chipsets to require as little change as possible.

SDN dates even back with ITU's SS7 (ITU 1993) networks where the signaling of telephone calls was separated from the actual phone call in order to setup and tear down phone calls, but that enabled new services to be formed such as local number portability and number translations. In addition ITU's ATM technology (ITU 1990) has been based on the concept of separating signaling and datapath, the signaling being used to set up the connections.

Next came the era of Active and Programmable Networks (A&PN), as surveyed by Tennenhouse et al. (1997) and Campbell et al. (1999) where network programmability was the focus. A&PN was based upon on a richer model than programmability and presented two alternatives: the in-band and out-of-band control.

In-band control was the most representative approach of the active networking school of thought from those years, where the concept was that code was actually traversing the network alongside the packets and was executed at specific nodes in the network such as Active Node Transfer System (ANTS) discussed by Wetherall et al. (1998). However it was the out-of-band control, the programmable networks approach, which dominated the research results and the experimentations that were carried out. The programmable networks concept was to allow software to control how the devices manipulate packets, the exact concept that current SDN proponents are advocating for.

There is a couple of interesting research projects that came out of the era of A&PN such as the P1520 (Biswas et al 1998), and Tempest (Rooney et al. 1998) projects. The IEEE P1520 standardization effort addressed the need for a set of standard software interfaces for programming networks in terms of rapid service creation and open signaling by defining a set of levels of abstraction and their respective interfaces similar to SDN concepts. The Tempest project, taking a cue from the advances of network virtualization