

Advances in Networking Software – Guest Editorial

Networking and communications systems are undergoing a substantive transformation on several fronts. These innovations portend substantially lower cost, simplified operations, and dramatically faster innovation cycles as traditional barriers to the deployment of innovations are removed. Where in the past networking functions were predominantly implemented using purpose-built hardware, custom protocols, and firmware images, those networking functions are increasingly instantiated through software that is abstracted from hardware, that would not be restricted to run only as part of a firmware image, freely programmable, and relying on algorithmic invocation of generic APIs. This transformation is best summarized as “softwarization” of the network, which is, in turn, realized through advances in networking software. These advances are multifaceted and occur in many areas, ranging from virtualization of networking functions to new network deployment models that allow for logical centralization, distribution, network slicing and efficient-placement of control function, from greater programmability and extensibility of networking devices to the re-emergence of programmable networks and over-the-top services, from new network interfaces to open source platforms and development toolkits, and more.

This Feature Topic features six articles that are exemplary of this transformation, providing an excellent cross-section across these facets. First *“NFV Orchestration Framework Addressing SFC Challenges”* by Mehtri Marouen, Ghribi Chaima, SOUALAH Oussama, Zeghlache Djamel presents an end-end-end framework that is geared towards the orchestration of service function chains involving virtualized network functions. The framework provides a great example of work that uses conceptually centralized platforms for control and orchestration that leverage both global network visibility as well as new interfaces, in this case, to control virtualized functions and compose them into service function chains to provide networking services.

While many efforts in network softwarization and specifically in Software-Defined Networks are geared towards extracting intelligence from networks to centralize control functions that were formerly distributed, other approaches are emerging that aim to move certain functions back towards the network edge. Placing functions at the edge can have advantages with regards to scaling as well as performance of locally closed control loops; it can be particularly attractive for applications that require only limited coordination and visibility that does not extend beyond the edge. This theme is exemplified in *“Container Network Functions: Bringing NFV to the Network Edge”* by Cziva Richard, Pezaros Dimitrios. This paper also highlights the rise of containers as a virtualization technology that has considerable advantages over traditional VMs in many deployment scenarios.

The third article, *“Programmable Overlays via OpenOverlayRouter”* by Rodriguez-Natal, Jordi Paillisse, Florin Coras, Albert Lopez-Bresco, Lorand Jakab, Marc Portoles-Comeras, Vina Ermagan, David Meyer, Dino Farinacci, Fabio Maino, Albert Cabellos-Aparicio, turns our attention towards advances in networking software with regards to the ability to program overlays on top of existing networks. Here the authors leverage LISP, a technology used to decouple the concept of identifiers to uniquely identify a system from the concept of locators used for purposes of routing, and introduce an open source platform with the purpose of programming LISP-based overlays.

Advances in networking software provide developers with great power to program network behavior, but with great power comes great responsibility. One aspect that developers need to confront is how to deal with rainy-day scenarios, exceptions, and race conditions. In this regard, *“Garbage Collection of Forwarding Rules in Software-Defined Networks”* by UI Huque Tanvir, Jourjon Guillaume, Gramoli Vincent presents a system that deals with one such aspect, namely the impact of reprogramming of forwarding rules in a network. The article describes implications of reprogramming flow tables in a software-defined network with regards to race conditions in the forwarding of packets that are in transit and the considerations required to conduct orderly cleanup and removal of prior forwarding rules.

Next, *“NEAT: A Platform- and Protocol-Independent Internet Transport API”* by Khademi, Khademi Naeem, Ros David, Welzl Michael, Bozakov Zdravko, Brunstrom Anna, Fairhurst Gorry, Grinnemo Karl-Johan Hurtig Per, Jones Tom, Mangiante Simone, Tuexen Michael, Weinrank Felix covers a different dimension of networking software, namely advances in the APIs used by applications to interact with network services. The authors present a transport API that can support different transports in a way that is transparent to applications, providing an alternative to socket programming.

Finally, *“ARPPIM: IP Address Resource Pooling and Intelligent Management System for Broadband IP Networks”* by Xie Chongfeng, Bi Jun, Yu Heng, Li Chen, Sun Chen, Liu Qing, Zheng Zhilong, Liu Shucheng illuminates advances in networking software as it relates to automation of management tasks and their migration from Operations Support Systems into SDN Controllers, in this case for the purposes of management of IP address assignments.

We believe that this unique combination of articles brings across a sense of the wide spectrum of advances we are currently witnessing in this field – it is not just SDN, not just NFV, but a much broader scope that is engulfing our industry. We hope that you will enjoy this Feature Topic and find these articles as inspirational as we do.