# SDN and NFV for Network Cloud Computing: a Universal Operating System for SD Infrastructures

Antonio Manzalini [1]
[1] Telecom Italia, Strategy and Innovation - Future Centre, via Reiss Romoli, 274 Torino, Italy

Noel Crespi [2]
[2] Institut Mines-Telecom, Telecom Sud Paris CNRS UMR5157, Paris, France

*Abstract— This paper proposes the medium-long term vision where Telecommunications networks will evolve towards becoming Software Defined Infrastructures (SDI). In particular, SDI will be highly dynamic and pervasive environments of logical resources capable of executing any network and service function (e.g., from L2 to L7) developed and controlled as software applications. It is argued that SDI will be a first concrete impact of the Softwarization trend which is starting to impact (with different expressions, e.g., SDN, NFV, Cloud and Network Computing) the Telecommunications and ICT ecosystems.*

*In this respect, the paper proposes the model of the Universal Operating System (UOS) for SDIs as an overarching and distributed Operating System spanning from terminals through the network elements, to the Cloud/IT resources. Then UOS functional architectures and some key challenges related to its development are briefly examined.*

*Eventually, it is argued that this evolution towards SDI will definitely blur, on one side the border between the Telecommunications networks and the Cloud Computing, and, on the other side, the distinction between the Telecommunications networks and the future "terminals" connected to them (e.g., handsets, tablets, machines, smart things, drones, robot). This transformation will have profound techno-economic implications for the Telecommunications and ICT ecosystems.*

***Keywords— Software Defined Infrastructures, SDN, NFV, Cloud Computing, Operating System, Softwarization, Future Networks and Services***

## I. Background and Context

Pervasive diffusion of ultra-broadband, increasing performance of chipsets, the tumbling costs of IT hardware and growing availability of open source software/hardware (e.g., in terms of applications, tools and platforms) are creating the conditions for a change of paradigm in exploiting and operating future Telecommunications infrastructures and services.

In this context, Software-Defined Network (SDN) [1] and Network Functions Virtualization (NFV) [2] are just two of the different facets of an overall systemic and disruptive transformation, called Softwarization of Telecommunications, ICT and other related ecosystems.

In fact, even if SDN and NFV are well-known paradigms, since a few decades, only today they are becoming really competitive (e.g., from the performances perspective) and sustainable (e.g., from the economic viewpoint). Cloud, Edge and Fog Computing and Networking are just other facets aspects of the same transformation: in the medium-long term all network functions and services of Telecommunications will be virtualized, dynamically allocated and executed as "applications" onto a hosting physical infrastructure, which will be fully decoupled from above software horizontal platforms.

Software-hardware decoupling and resources virtualization will pave the way towards Software Defined Infrastructures (SDI), capable of breaking the borders between the Cloud, the network and the sheer number of smartphones, interconnected machines, devices and smart things (e.g., Internet of Things).

This is an impactful change of paradigm with respect to the past when Telecommunications network infrastructure have been built with purpose-built equipment designed for specific functions; these pieces of equipment were provided by Vendors as "closed boxes" including the hardware, software and its operating system, bringing this to the development of several vertical "silos" in legacy Telecommunications infrastructures.

As mentioned, SDI paradigm is based on the concept of breaking network and service functions and developing them as applications. This paradigm is not limited to decoupling software components from the hardware equipment, but it relies on a modularization of CPU, memory and network capabilities. Then these modules will be abstracted and virtually aggregated to meet application demands. This is why the SDI paradigm overcomes the "ossification" of several vertical "silos" in legacy Telecommunications infrastructures.

This innovation, if properly exploited, will bring dramatic costs reductions (both in CAPEX and OPEX) and it will open new socio-economic and business opportunities. Let's consider, for example the combination of mass digitalization and the introduction of the "machine intelligence" (e.g. mobile robots, drones and other forms of autonomous machines) into any processes of industries, agriculture, public institutions, etc. This will enable the true impending arrival of the Digital Society and the Digital Economy.

It should be also noted that this evolution is lowering "thresholds" for new Players to enter the telecommunications and ICT markets [3], as less investments are required: this orientation of the business towards OPEX-centric models will create new roles, relationships and changes in the value chains. As such, Softwarization is a very complex transformation (it is not just about the introduction of another overlay technology), and, more importantly, it concerns not only the technological aspects but more importantly also socio-economic, cultural ones and strategic regulatory rules.

In short, it is argued that in the medium-long term Telecommunications infrastructures will evolve towards becoming SDIs. SDIs will look like highly distributed Network and Cloud Computing platforms capable of executing any network functions and services virtualised as software applications [4]. This will definitely blur, on one side the separation between the Networks and the Cloud Computing, and, on the other side, the distinction between the Networks and the future "terminals" connected to them (i.e., handsets, tablets, machines, smart things, drones, robot). It is likely that this wave of innovation will bring first concrete exploitations by 2020, through the deployment of 5G [5], a first expression of a SDI.

SDIs will look like pervasive, massive, multi-domain like highly distributed Network and Cloud Computing platforms, i.e., sort of "Supercomputers", with fixed and mobile ultra-low latency links providing the pervasive connectivity. If so, then we can imagine SDIs requiring a sort of Universal Operating System (UOS). UOS, for example, should be in charge of managing the hardware and software resources and providing sets of services, programmable with APIs. On top of this UOS, network and service applications will flourish to create and develop the new ICT ecosystems for the Digital Society and Digital Economy.

It can be said also that with SDIs the economic value is moving from hardware to software, and that the software will be the mean for implementing flexible architectures-of-architectures based on cognitive, deep-learning algorithms, A.I. methods...crunching datasets to infer decisions, to learn and to actuate actions onto the reality.

Main systemic characteristics should be universality (e.g., based on recursivness of a limited number of functional component and blocks), horizontality and openness (e.g. adopting open source software and hardware).

There is a number of challenges which have to be faced to bring this vision into reality. Some of them are:

- what are the abstractions to be provided and used at the different levels of a SDI ?
- what virtualization techniques should be adopted for the different types of resources: e.g., virtualized network functions can be implemented as a full VM using a server node, virtualization based on hypervisor (e.g. QEMU, Xen, KVM, and VMware), as a Linux Container, (e.g. LXC)...?
- how orchestrating applications life-cycles, how managing software and hardware infrastructure's resources ?
- what kind of controllers adopting for different kind resources (up to terminals) of a SDI ?
- which levels of "programmability" will be offered to Third Parties and Users through dynamic APIs ?
- how automating processes: eTOM vs approaches *a la* DevOps ?
- how "mitigating" CAP [8] Theorem limitations ?
- how reducing the end-to-end latency in a SDI up to units of milliseconds? This is key factor of success for deploying new service models (e.g., robot-as-a-service, immersive communications, etc);
- how ensuring multi-domain, multi-vendor interoperability for virtual platforms (mind the clash Standardization traditional processes vs Standards-de-facto);
- how facing Security and Privacy issues with solutions "by design" ?

The above list can be considered a sort of research agenda for the development and exploitation of SDIs.

The outline of this paper is as follows: section II elaborates about the Softwarization trends impacting Telecommunications and ICT and presents some future scenarios by introducing the vision of SDIs; section III describes the three layers functional architecture of an SDI, addressing some of the key challenges; section IV elaborates the need of developing UOS with open source software components, motivating this is a key factor of success for the whole Telecommunications and ICT ecosystems. Eventually closing remark are provided in the last section.

## II. Future Telecommunications and ICT Scenarios

It has been mentioned that Cloud-, Edge- and Fog Computing, SDN and NFV can be seen as different facets of this same evolutionary trend, which is called "Softwarization". This trend is likely to have huge industrial and socio-economic impacts and implications, creating both risks and opportunities for Telecommunications Operators, Service Providers, and, in general, for all the Industries embracing (at what level) this wave of innovation.

Let's consider, as an example, a long term scenario where Operators will become fully Virtual Operators by owning and operating SDIs. Considering the acceleration of IT advances, this might be possible (at least to a certain extent) already with the advent of 5G (e.g. by 2020): this, clearly, will depend on a number of factors, including the development of models of business sustainability and a proper regulation rules. These advanced Virtual Operators (which we may call Software Defined Operators) could "borrow" physical and hardware raw resources (e.g. just antennas, L0-L1 transmission and processing and storage capabilities) from some Infrastructure Providers. In this case they will become flexible and fastacting (as OTTs) on a global markets to provide Users and Enterprises with "programmable" (with dynamic APIs) and "de-perimetrised" ICT services. This is potentially a game changer for all the Telecommunications industries, creating risks but also new business opportunities and, potentially, a new socio-economic development for the Digital Society.

In general, Service Providers and Network Operators fully embracing this innovation wave will see dramatic costs reductions (e.g., estimation of 40%-50% CAPEX savings), improved efficiency in the overall Operations (e.g., estimation of 25%-35% OPEX savings only by automating processes), reduced time-to-market, greater flexibility and adaptability to new emerging service paradigms. The risk is the temptation of integrating this innovative approaches into legacy infrastructures (Red Ocean strategy), accumulating delays (due the complication of this approach) and thus being surpassed by Players faster in adopting a Blue Ocean strategy in deploying SDIs.

Being the Telecommunications infrastructures fully dematerialised, also other disruptive scenarios will be possible. Software Defined Operators will be also potentially capable to "upload and execute" their "software networks and services platforms" anywhere there will be an Infrastructure Providers willing to rent hardware and L0-L1 resources.

Costs saving (Capex and Opex) is not enough for Telecommunications business sustainability. New source of revenues should be created through new service paradigms.

SDIs will enable new paradigms, such as "Anything as a Service": intelligent terminals, devices and machines, robots will allow, for example, improving industrial and agricultural efficiency, developing new models of decentralized micro-manufacturing [6] (e.g. also through 3D printers), improving efficiency in public processes, saving energy, supporting better ageing citizens' lives (e.g. with machines and robots), providing a richer education (for all at lower costs), fully "digitalizing" the Society and enabling new sustainable ICT ecosystems. Also "Immersive Communications" will be enabled by SDIs: beyond the "commoditization" of current communication paradigms (e.g., voice, messaging, etc), new forms of communications (e.g., artificially intelligent avatars, cognitive robots-humans interactions, etc) will require ultra-low latency and highly flexible platforms as the ones of SDIs.

Next sub-section will describe some examples of foreseen service scenarios, enabled by SDIs.

*1)* SDI enabling the "Second Machine Age"

The number of terminals, intelligent machines and smart things (which are embedding communications, sensors and actuators) "attached" to current Telecommunications networks is growing exponentially. It can be argued that the Softwarization of Telecommunications networks will contribute to increase even more this phenomenon in the future, paving the way to the so-called "Second Machine Age" [11].

This trend will bring, in the medium-long term, a number of socio-economics consequences, for example: reduction of human manual efforts in repetitive or dangerous jobs (at least those which are subjected to computerization and robotization); increase of local production and as such also reduction of long distance transportation; "optimization" in electrical power production and consumption, etc. Eventually this will provide higher quality of life, cheaper goods and better services for consumers.

As a matter of fact, industry, agriculture, mining, security, and other infrastructure companies are already expressing strong interest in using robots and autonomous machines. In agriculture, for example, for tasks like crop inspection, targeted use of water and pesticides, actions and monitoring to assist farmers, as well as in data gathering for optimising the processes. In this context, SDI can operate in real time autonomous machines (as they were nodes) for a number of different agricultural applications. Interestingly, also in this context, APIs can be opened to end-Users and Third Parties to develop, program and provide any related services and applications.

In general robotics applications which will make our spaces more comfortable, safer and functional also in our third and fourth age. It is estimated that by 2050-2060 the European demographics will be represented for about one third by people over 65. There will be only two people in the working age for each elderly. The cost of the combined pension and health care system will top about 29% of the European GDP.

Given these numbers, it is expected that robotics would have an impact in helping this large section of the population to stay active and independent, increase their quality of life and maintain social cohesion. SDI operated autonomous machines and robots will enable remote medicine and open up a new world of domestic applications which may be taken uniformly by the entire population (e.g. cleaning, cooking, playing, communicating, etc.).

As another example, SDI can contribute in progressing the digitalization of enterprises so to create global digital environments where enterprises can play multiple roles cooperating and competing in global markets in a more effective, dynamic and flexible way. One can image feedback mechanisms (based on almost real-time big data processing) to improve performance and productivity of processes by self-corrections of (internal and external) actions. All of this by sharing the knowledge created and mediated by an highly flexible and ultra-low latency SDI offering huge amount of processing and storage power. In fact, enterprises processes can be mapped in huge data sets, which can be processed with big data analytics and cognition methods to actuate then optimized operations. This will pave the way to the Industry 4.0 paradigm.

It is argued that these scenarios will be enabled by the SDIs characteristics: highly distributed Network and Cloud Computing platforms with fixed and mobile ultra-low latency links providing pervasive connectivity to a massive "fabric" of processing and storage resources.

## III. SDI ARCHECTURE AND RELATED CHALLENGES

Figure 1 represents the functional architecture of a SDI. Specifically the architecture is structured into three main layers (i.e., Application Layer, Service Layer and UOS Layer) hosted on top of a physical infrastructure, which includes terminals (even the more advanced ones, e.g., intelligent machines, smart things, robots, drones), data plane network elements (e.g., switches, routers) and Cloud/IT physical systems (e.g., CPU and memory/storage resources).

Applications belong to Application Layer, which, uses centralised (e.g. Cloud Computing) and distributed (e.g., network nodes, mini Data Centres and terminals) processing and storage capabilities. In particular, from a functional view point, the applications interact (via APIs) with the underneath Service Layer by sending so-called service requests.

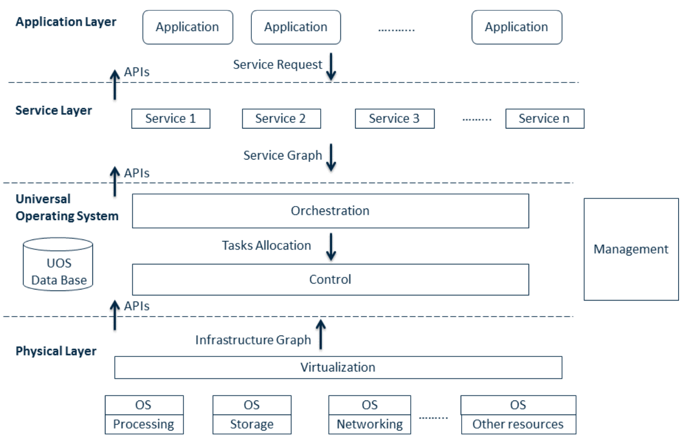


Figure 1 - Functional architecture of a SDI

The Service Layer is in charge of providing the Application Layer with all the required service components. In fact, upon receiving the service request, the Service Layer is generating a service graph which is a logical decomposition of the service request in a sequence of tasks (e.g., Virtual Network Functions), with the related QoS requirements. Then, tasks of the service graph have to be properly allocated for execution onto the physical infrastructure.

The Universal Operating System (UOS) is linking the Service Layer to the physical infrastructure. UOS is an overarching and distributed Operating System spanning horizontally from terminals, through the network elements, to the Cloud/IT resources. The distributed architecture has obvious advantages such as modularity, integration capability and scalability. UOS should be as lightweight/minimal as possible to achieve great flexibility and evolution. U-OS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.

Specifically, one of the main tasks of the UOS concerns the creation of a virtual infrastructure (composed by logical resources), on top of which instantiating, configuring and starting/stopping the necessary software tasks. This is very crucial in future SDI, where processing resources will be highly distributed (i.e., up to the local terminals): in fact a key challenge is allocating the tasks whilst minimizing the overall end-to-end latency.

Optimized allocation and move [7] of network and service functions across wide area, will make SDI resilient against failure, disaster and traffic dynamics. Allocation, in fact, comprises the selection of the resource instances with respect to the deployment variants, the capability of the interfaces and the logical as well as the physical topology. This selection should be done in a way to address specified service requirements (e.g., minimization of the overall latency).

The dynamic allocation and orchestration of tasks will be even more challenging when considering also the heterogeneity of SDI resources. In fact, UOS should have the capability to cope with the heterogeneity of the underlying future SDI, not only in terms of operating systems (e.g., Android, Robot Operating System, OpenStack, ON.OS, etc) but also in terms of hardware (the edge of the network will be mostly in the hands of MIPS and ARM, while the core will have x-86 plus possibly dedicated hardware for specific functions).

It should be also possible programming, allocating and moving a variety of virtual architectures on-demand, based on Users' requests, governance and business requirements overcoming the current ossified networks structures. End Users will have access to a certain number of abstraction for programming, setting-up and tearing down, migrate and optimize their network functions and services (e.g. local traffic engineering, failure handling policy, local topology optimization, etc.) according to their need and service level agreement.

Legacy OSS/BSS systems (mostly still based on manual configurations) will not be able cope with this dramatic growth in complexity. New cognitive and automated methods and systems for management, tightly integrated with the UOS, will be required. In the long term, a key challenge will be "operating" dynamically millions of software processes that will be continuously executed, in scenarios where machines, robots, drones and smart things will become dominant Users.

Moreover UOS should have an up-to-date view of the resources states (i.e., big data) and it should be able to process them in a meaningful way. First of all it is necessary to retrieve up-to-date and consistent data describing the resources with the required level of detail. To this end a data semantic, syntax of the communication method, filtering and storage and strategy are also necessary.

The physical layer includes all the physical resources composing the SDI. In particular, the infrastructure graph (figure 2) gives a representation of the physical topology and provides information about the status of the SDI elements. The infrastructure graph has basic objects like router, switch, port, hosts, devices and links. It can be said that Applications programs the data plane (represented by the infrastructure graph) according to the service graph.

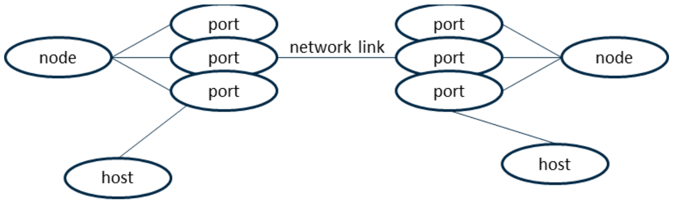


Figure 2 – Example of infrastructure graph

A SDI should also have a catalog of the consistency models. Concerning the consistency models of the various states, more research is needed. As known, the CAP theorem [8] states that any networked shared-data system can have at most two of following three properties: 1) Consistency (C) equivalent to having a single up-to-date copy of the data; 2) high Availability (A) of that data (for updates); and 3) tolerance to network Partitions (P).

Considering a SDI as a distributed computational system (routing/forwarding packets can be seen as a computational tasks being protocols themselves distributed solutions of constraints optimization problems), just two of the three CAP properties will be possible at the same time. The general idea is that 2 of the 3 have to be chosen. CP favor consistency, AP favor availability, CA there are no partition.

This has profound implications for technologies that need to be developed in line with the "deploy with repeatable, reliable processes" principle for configuring the states of a SDI. Latency or delay and partitioning properties are deeply related, and such relation becomes more important in the case of telecom service providers where Devs and Ops interact with widely distributed infrastructure. Limitations of interactions between centralized management and distributed control need to be carefully examined in such environments. Traditionally connectivity was the main concern: C and A was about delivering packets to destination. The features and capabilities of SDN and NFV are changing the concerns: for example in SDN, control plane Partitions no longer imply data plane Partitions, so A does not imply C. In practice, CAP reflects the need for a balance between local/distributed operations and a remote/centralized operations.

## IV. UOS Development with Open Source Software

In the last section, it has been argued that UOS should be highly distributed and horizontally scalable, it should be highly available and rely on the physical infrastructure abstractions.

Concerning UOS development, it is argued that it should be built with open source components. This is a key factor of success for the whole Telecommunications ecosystem. One may ask then: which is the best to achieve that ?

Looking at the past, there have been some examples of successful development and adoption of Open Source software, producing a strong impact on the market, even in short span of time (e.g., less than 5 years).

Linux kernel is one prominent example: it is a Unix-like operating system released (around 1994) under the GNU General Public License version (GPLv2). The Linux kernel is used by a variety of operating systems based on it, which are usually in the form of Linux distributions. An example of "good enough" but simpler and cheaper operating system. The popularity of Linux virally diffused across developers and users. Linux servers now represent about 29% of all server revenue (which means, given their lower costs with respect to other commercial servers, a much larger market share).

Apache is another example. Apache has been the most popular web server on the Internet since April 1996. Apache adoption diffused virally reaching about 60% of the market in about 5 years. Today, 385 million sites are now powered by Apache, landing today on a 38% share of the market.

Concerning open source OS for smartphone, in 2014 Android rose to about 85% of the global market share. As known, most of Android is free and open source, even if a large amount of software on Android devices (such as Play Store, Google Search, Google Play Services, Google Music, and so on) are proprietary and licensed. Google does not charge a fee for companies that install Android on their devices. Nevertheless every installed copy of Android gives Google potential customers for its services and encourages Developers to build apps for the platform, this creating ecosystems around Android.

Coming to Cloud, SDN, NFV, there are several other recent examples. OpenStack [9] is an open source platforms that can be used to provide Cloud services. OpenStack has been developed by the OpenStack Foundation, now comprising a large group of technology companies. Several companies in the Foundation run public cloud services on the OpenStack platform (comparable with Amazon's Elastic Computing Cloud EC2). OpenStack has been launched in 2010 and today dominates the market with a 69% adoption rate of IT Enterprises offering Cloud Computing services.

No need mentioning the long list of open source SDN controllers which have been released and are being tested. Moreover, ON.LAB has released in open source an operating systems (ON.OS) [10] for SDN-like networks: interestingly, after the release some Telco Network and Service Providers immediately announced that they will start testing it for assessing a future potential deployment in production environments.

It is argued that UOS developments should leverage on this large set of open source pieces of software, with a systemic integration effort starting from a shared SDI functional model.

The lesson learnt from the past is that if an open source software solution will be developed and released with the support of a critical mass of companies, it is likely that in about 5 years it will reach at least 50%-60% adoption in the related market. This is a reasonable expectation also for the UOS.

## V. Concluding Remarks

Softwarization is going to have a huge impact in the Telecommunications and ICT domains. Service Providers and

Virtual Operators fully embracing this innovation wave may see dramatic costs reductions (e.g., estimation of 40%-50% CAPEX savings), improved efficiency in the overall Operations (e.g., estimation of 25%-35% OPEX savings only by automating processes), reduced time-to-market, greater flexibility and adaptability to new emerging service paradigms. Telecommunications and ICT business ecosystems will dramatically change.

It can be argued that Softwarization will enable the so-called "Second Machine Age" [11] bringing, in the medium-long terms, a number of socio-economics consequences: reduction of human manual efforts in jobs (e.g., subjected to computerization, robotization); increase of local production; reduction of long distance transportation; "optimization" in electrical power production and consumption. On a short-term basis, this transformation will create a ripple on the current socio-economic ecosystems. This increased automation will provide better quality and cheaper goods and services for consumers.

In this scenario, the paper proposed the vision of SDI as a massively dense distributed computational and communication infrastructure entering deeply into our socio-economic reality. In particular, just as a computer has an operating system — dictating the way it works and provides services as a foundation upon which all applications are built — this paper argued that SDI should have a Universal Operating System.

In SDI, the distinction between the Network and the Cloud is likely to disappear, as more and more functionalities will be performed either in the Network or in the Cloud depending on the performance requirements and costs optimizations. The control and orchestration capabilities will be the key factor of success in taming the complexity of an infrastructure executing millions of software transactions (or service chains).

This should reflects also the way in which "innovation" is being developed and exploited. In fact, the past "waterfall" model of making innovation is going to be finished: that model was moving from research activities to standardization, from systems development by Technology Providers to Service Providers deployments up to services provisioning to Consumers. Already today (and more and more tomorrow) innovation is likely working the other way round: it is starting from the combination of the needs of the Digital Society, from the requests of massive Consumers, Users and from the challenging cross-requirements of the new Digital Economy. This has been referred as "Complex Innovation" [12].

In this sense, an open collaboration between a critical mass of different Players, which could contribute to build this vision, is required. A first milestone could be developing standard functional architecture of SDI and the related requirements; then mainly involving (and funding) open source communities and initiatives to develop the SDI software. Integration will be the last phase, carried out case-by-case.

Finally, because of this evolution, several economists, as well as technologists, have started to wonder if the usual representation of relationships among a myriad of players in a certain industrial area can still be modeled on the bases of value chains. There is a growing consensus that value chains modeling shall be complemented by a broader view considering business ecosystems. Current regulation should evolve to support this digital economy, making it sustainable.

## REFERENCES


[1] White paper on "Software-Defined Networking: The New Norm for Networks" https://www.opennetworking.org/;

[2] White paper on "Network Functions Virtualisation" http://portal.etsi.org/NFV/NFV_White_Paper.pdf;

[3] A. Manzalini, R. Saracco, "Software Networks at the Edge: a shift of paradigm", IEEE Workshop SDN4FNS, Trento, November 2013;

[4] A. Manzalini, R. Saracco, C. Buyukkoc, P. Chemouil, et al, "Software-Defined Networks for Future Networks and Services: main technical challenges and business implications," IEEE SDN4FNS, White Paper, January 2014;

[5] David Soldani, Antonio Manzalini , "5G: The Nervous System of the True Digital Society" IEEE COMSOC MMTC E-Letter, September 2014;

[6] A. Manzalini, A. Stavdas, "The Network is the Robot", Communication and Strategies Communications & Strategies, ERN Economics of Networks eJournal, Dossier to be published in No. 96 - 4th quarter 2014;

[7] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, "Clouds of Virtual Machines at the Edge", IEEE Com. Mag. "Future Carriers Networks", July 2013;

[8] E. Brewer, "Towards Robust Distributed Systems," Proc. 19th Ann. ACM Symp.Principles of Distributed Computing *(PODC* 00), ACM, 2000;

[9] Openstack: http://www.openstack.org/;

[10] ON.OS: http://onosproject.org/;

[11] E. Brynjolfsson and A. McAfee, "The Second Machine Age", W.W. Norton & Company, New York, 2014, London;

[12] http://ieee-sdn.blogspot.it/2014/07/from-waterfall-towards-complex.html.