NETWORKS & TELECOMMUNICATIONS SERIES

ADVANCED NETWORKS SET

Volume 1

# Software Networks

*Virtualization, SDN, 5G and Security*

**Guy Pujolle**

ISTE

WILEY

Software Networks

**Advanced Networks Set**

coordinated by
Guy Pujolle

Volume 1

# Software Networks

*Virtualization, SDN, 5G and Security*

Guy Pujolle

ISTE

WILEY

First published 2015 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.



ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com



Library of Congress Control Number: 2015942608

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-84821-694-5

# Contents

# Introduction

Currently, networking technology is experiencing its third major wave of revolution. The first was the move from circuit-switched mode to packet-switched mode, and the second from hardwired to wireless mode. The third revolution, which we examine in this book, is the move from hardware to software mode. Let us briefly examine these three revolutions, before focusing more particularly on the third, which will be studied in detail in this book.

## I.1. The first two revolutions

A circuit is a collection of hardware and software elements, allocated to two users – one at each end of the circuit. The resources of that circuit belong exclusively to those two users; nobody else can use them. In particular, this mode has been used in the context of the public switched telephone network (PSTN). Indeed, telephone voice communication is a continuous application for which circuits are very appropriate.

A major change in traffic patterns brought about the first great revolution in the world of networks, pertaining to asynchronous and non-uniform applications. The data transported for these applications make only very incomplete use of circuits, but are appropriate for packet-switched mode. When a message needs to be sent from a

transmitter to a receiver, the data for transmission are grouped together in one or more packets, depending on the total size of the message. For a short message, a single packet may be sufficient; however, for a long message, several packets are needed. The packets then pass through intermediary transfer nodes between the transmitter and the receiver, and ultimately make their way to the end-point. The resources needed to handle the packets include memories, links between the nodes and sender/receiver. These resources are shared between all users. Packet-switched mode requires a physical architecture and protocols – i.e. rules – to achieve end-to-end communication. Many different architectural arrangements have been proposed, using protocol layers and associated algorithms. In the early days, each hardware manufacturer had their own architecture (e.g. SNA, DNA, DecNet, etc.). Then, the OSI model (Open System Interconnection) was introduced in an attempt to make all these different architectures mutually compatible. The failure of compatibility between hardware manufacturers, even with a common model, led to the re-adoption of one of the very first architectures introduced for packet-switched mode: TCP/IP (Transport Control Protocol/Internet Protocol).

The second revolution was the switch from hardwired mode to wireless mode. Figure I.1 shows that, by 2020, terminal connection should be essentially wireless, established using Wi-Fi technology, including 3G/4G/5G technology. In fact, increasingly, the two techniques are used together, as they are becoming mutually complimentary rather than representing competition for one another. In addition, when we look at the curve shown in Figure I.2, plotting worldwide user demand against the growth of what 3G/4G/5G technology is capable of delivering, we see that the gap is so significant that only Wi-Fi technology is capable of handling the demand. We shall come back to wireless architectures, because the third revolution also has a significant impact on this transition toward radio-based technologies.

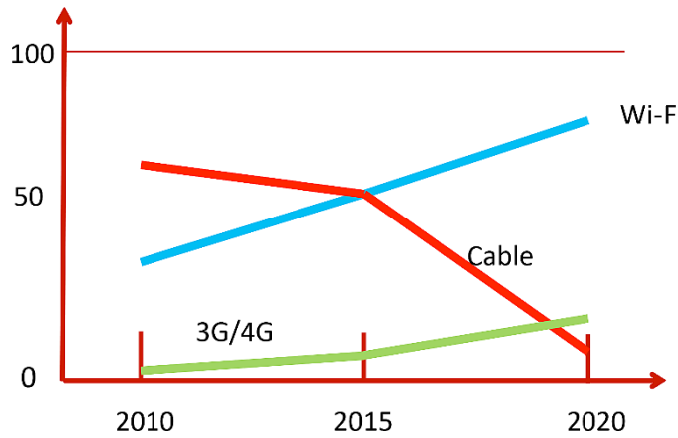



**Figure I.1.** *Terminal connection by 2020*

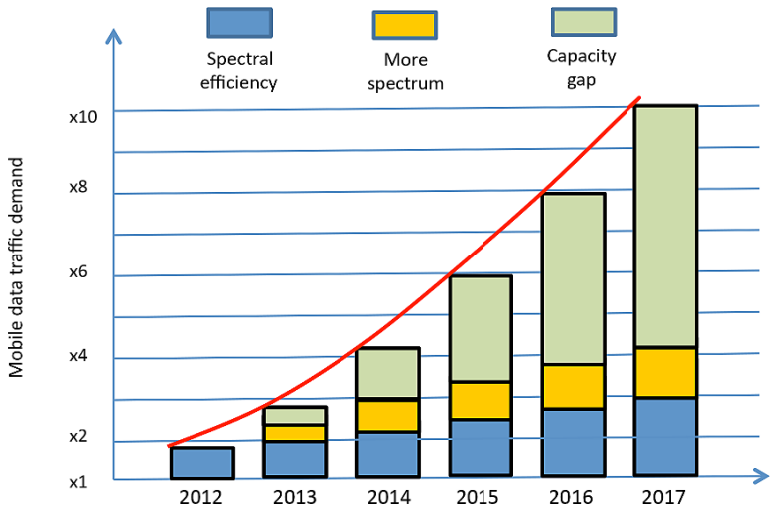



**Figure I.2.** *The gap between technological progress and user demand. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

## I.2. The third revolution

The third revolution, which is our focus in this book, pertains to the move from hardware-based mode to software-based mode. This transition is taking place because of virtualization, whereby physical networking equipment is replaced by software fulfilling the same function.

Let us take a look at the various elements which are creating a new generation of networks. To begin with, we can cite the Cloud. The Cloud is a set of resources which, instead of being held at the premises of a particular company or individual, are hosted on the Internet. The resources are de-localized, and brought together in resource centers, known as datacenters.

The reasons for the Cloud's creation stem from the low degree of use of server resources worldwide: only 10% of servers' capacities is actually being used. This low value derived from the fact that servers are hardly used at all at night-time, and see relatively little use outside of peak hours, which represent no more than 4-5 hours each day. In addition, the relatively-low cost of hardware meant that, generally, servers were greatly oversized. Another factor which needs to be taken into account is the rising cost of personnel to manage and control the resources. In order to optimize the cost both of resources and engineers, those resources need to be shared. The purpose of Clouds is to facilitate such sharing in an efficient manner.

Figure I.3 shows the growth of the public Cloud services market. Certainly, that growth is impressive, but in the final analysis, it is relatively low in comparison to what it could have been if there were no problems of security. Indeed, as the security of the data uploaded to such systems is rather lax, there has been a massive increase in private Clouds, taking the place of public Cloud services. In Chapter 6, we shall examine the advances made in terms of security, with the advent of secure Clouds.

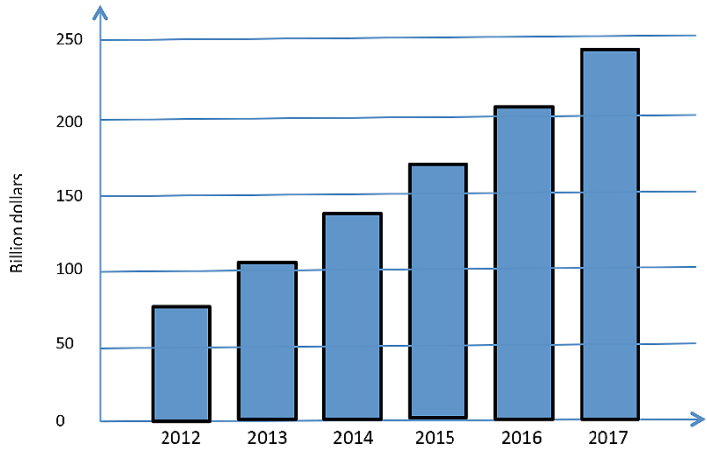



**Figure I.3.** *Public Cloud services market and their annual growth rate*

Virtualization is also a key factor, as indicated at the start of this chapter. The increase in the number of virtual machines in undeniable, and in 2015 more than two thirds of the servers available throughout the world are virtual machines. Physical machines are able to host increasing numbers of virtual machines. This trend is illustrated in Figure I.4. In 2015, each physical server hosts around eight virtual machines.

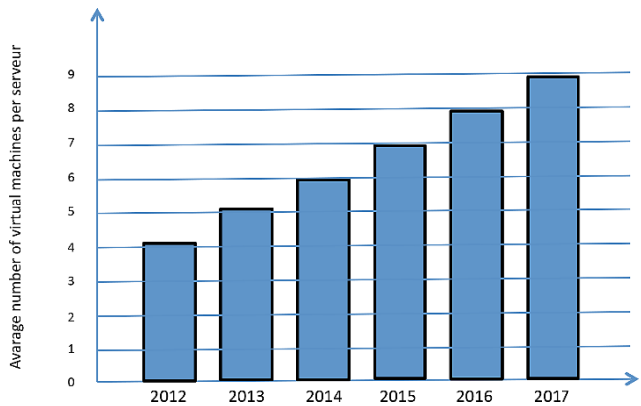



**Figure I.4.** *Number of virtual machines per physical server*

The use of Cloud services has meant a significant increase in the data rates being sent over the networks. Indeed, processing is now done centrally, and both the data and the signaling must be sent to the Cloud and then returned after processing. We can see this increase in data rate requirement by examining the market of Ethernet ports for datacenters. Figure I.5 plots shipments of 1 Gbps Ethernet ports against those of 10 Gbps ports. As we can see, 1 Gbps ports, which are already fairly fast, are being replaced by ports that are ten times more powerful.

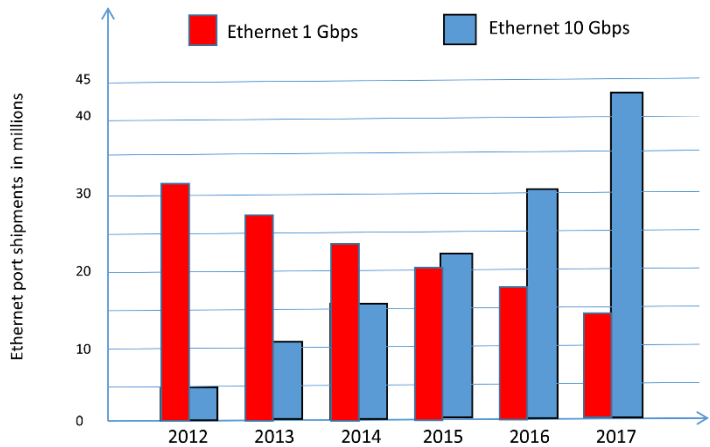



**Figure I.5.** *The rise in power of Ethernet ports for datacenters*

The world of the Cloud is, in fact, rather diverse, if we look at the number of functions which it can fulfill. There are numerous types of Clouds available, but three categories, which are indicated in Figure I.6, are sufficient to clearly differentiate them. The category which offers the greatest potential is the SaaS (Software as a Service) cloud. SaaS makes all services available to the user– processing, storage and networking. With this solution, a company asks its Cloud provider to supply all necessary applications. Indeed, the company subcontracts its IT system to the Cloud provider. With the second solution – PaaS (Platform as a Service) – the company remains responsible for the applications. The Cloud provider offers a complete platform, leaving only the management of the applications to the company. Finally, the third solution – IaaS (Infrastructure as a

Service) – leaves a great deal more initiative in the hands of the client company. The provider still offers the processing, storage and networking, but the client is still responsible for the applications and the environments necessary for those applications, such as the operating systems and databases.

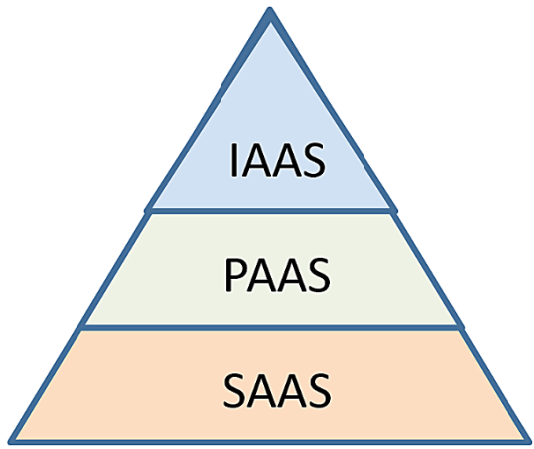



**Figure I.6.** *The three main types of Cloud*

More specifically, we can define the three Cloud architectures as follows.

– IaaS (Infrastructure as a Service): this is the very first approach, with a portion of the virtualization being handled by the Cloud, such as the network servers, the storage servers, and the network itself. The Internet network is used to host PABX-type machines, firewalls or storage servers, and more generally, the servers connected to the network infrastructure;

– PaaS (Platform as a Service): this is the second Cloud model whereby, in addition to the infrastructure, there is an intermediary software program corresponding to the Internet platform. The client company's own servers only handle the applications;

– SaaS (Software as a Service): with SaaS, in addition to the infrastructure and the platform, the Cloud provider actually provides the applications themselves. Ultimately, nothing is left to the

company, apart from the Internet ports. This solution, which is also called Cloud Computing, outsources almost all of the company's IT and networks.

Figure I.7 shows the functions of the different types of Cloud in comparison with the classical model in operation today.

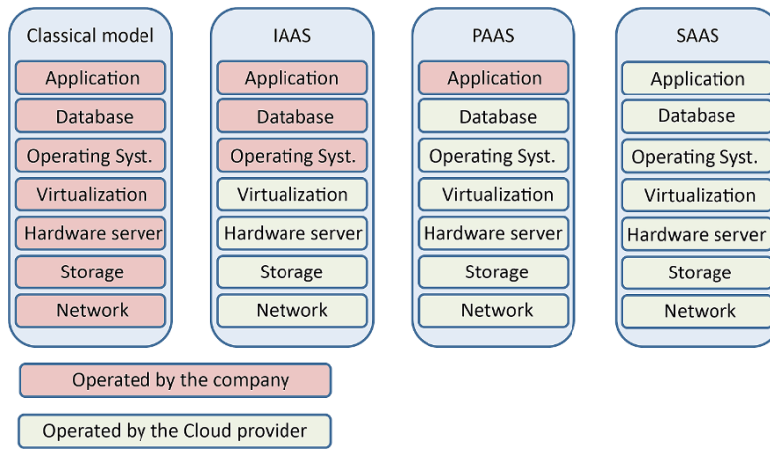



**Figure I.7.** *The different types of Clouds*

The main issue for a company that operates a Cloud is security. Indeed, there is nothing to prevent the Cloud provider from scrutinizing the data, or – as much more commonly happens – the data from being requisitioned by the countries in which the physical servers are located; the providers must comply. The rise of sovereign Clouds is also noteworthy: here, the data are not allowed to pass beyond the geographical borders. Most states insist on this for their own data.

The advantage of the Cloud lies in the power of the datacenters, which are able to handle a great many virtual machines and provide the power necessary for their execution. Multiplexing between a large number of users greatly decreases costs. Datacenters may also serve as hubs for software networks and host virtual machines to create such networks. For this reason, numerous telecommunications operators have set up companies which provide Cloud services for the operators themselves and also for their customers.

In the techniques which we shall examine in detail hereafter, we find SDN (Software-Defined Networking), whereby multiple forwarding tables are defined, and only datacenters have sufficient processing power to perform all the operations necessary to manage these tables. One of the problems is determining the necessary size of the datacenters, and where to build them. Very roughly, there are a whole range of sizes, from absolutely enormous datacenters, with a million servers, to femto-datacenters, with the equivalent of only a few servers, and everything in between.

## I.3. "Cloudification" of networks

The rise of this new generation of networks, based on datacenters, has an impact on energy consumption in the world of ICT. This consumption is estimated to account for between 3% and 5% of the total carbon footprint, depending on which study we consult. However, this proportion is increasing very quickly with the rapid rollout of datacenters and antennas for mobile networks. By way of example, a datacenter containing a million servers consumes approximately 100 MW. A Cloud provider with ten such datacenters would consume 1 GW, which is the equivalent of a sector in a nuclear power plant. This total number of servers has already been achieved or surpassed by ten well-known major companies. Similarly, the number of 2G/3G/4G antennas in the world is already more than 10 million. Given that, on average, consumption is 1500 W per antenna (2000 W for 3G/4G antennas but significantly less for 2G antennas), this represents around 15 GW worldwide.

Continuing in the same vein, the carbon footprint produced by energy consumption in the world of ICT is projected to reach 20% by 2025. Therefore, it is absolutely crucial to find solutions to offset this rise. We shall come back to this in the last chapter of this book, but there are solutions that already exist and are beginning to be used. Virtualization represents a good solution, whereby multiple virtual machines are hosted on a common physical machine, and a large number of servers are placed in standby mode (low power) when not in use. Processors also need to have the ability to drop to very low speeds of operation whenever necessary. Indeed, the power consumption is strongly proportional to processor speed. When the

processor has nothing to do, it almost stops, and then speeds up depending on the workload received.

Mobility is also another argument in favor of adopting a new form of network architecture. We can show that by 2020, 95% of devices will be connected to the network by a wireless solution. Therefore, we need to manage the mobility problem. Thus, the first order of business is management of multi-homing – i.e. being able to connect to several networks simultaneously. The word "multi-homing" stems from the fact that the terminal receives several IP addresses, assigned by the different connected networks. These multiple addresses are complex to manage, and the task requires specific characteristics. Mobility also involves managing simultaneous connections to several networks. On the basis of certain criteria (to be determined), the packets can be separated and sent via different networks. Thus, they need to be re-ordered when they arrive at their destination, which can cause numerous problems. Mobility also raises the issues of addressing and identification. If we use the IP address, it can be interpreted in two different ways: user identification enables us to determine who the user is, but an address is also required, to show where that user is. The difficulty lies in dealing with these two concepts simultaneously. Thus, when a customer moves sufficiently far to go beyond the sub-network with which he/she is registered, it is necessary to assign a new IP address to the device. This is fairly complex from the point of view of identification. One possible solution, as we can see, is to give two IP addresses to the same user: one reflecting his/her identity and the other the location.

Another revolution that is currently under way pertains to the "Internet of Things" (IoT): billions of things will be connected within the next few years. The prediction is that 50 billion will be connected to the IoT by 2020. In other words, the number of connections will likely increase tenfold in the space of only a few years. The "things" belong to a variety of domains: 1) domestic, with household electrical goods, home health care, home management, etc.; 2) medicine, with all sorts of sensors both on and in the body to measure, analyze and perform actions; 3) business, with light level sensors, temperature sensors, security sensors, etc. Numerous

problems arise in this new universe, such as identity management and the security of communications with the sensors. The price of identification is often set at $40 per object, which is absolutely incompatible with the cost of a sensor which is often less than $1. Security is also a complex factor, because the sensor has very little power, and is incapable of performing sufficiently-sophisticated encryption to ensure the confidentiality of the transmissions.

Finally, there is one last reason to favor migration to a new network: security. Security requires a precise view and understanding of the problems at hand, which range from physical security to computer security, with the need to lay contingency plans for attacks that are sometimes entirely unforeseeable. The world of the Internet today is like a bicycle tire which is now made up entirely of patches (having been punctured and repaired multiple times), and every time an attack succeeds, a new patch is added. Such a tire is still roadworthy at the moment, but there is the danger that it will burst if no new solution is envisaged in the next few years. At the end of this book, in Chapter 7, we shall look at the secure Cloud, whereby, in a datacenter, a whole set of solutions is built around specialized virtual machines to provide new elements, the aim of which is to enhance the security of the applications and networks.

An effective security mechanism must include a physical element: a safe box to protect the important elements of the arsenal, necessary to ensure confidentiality, authentication, etc. Software security is a reality, and to a large extent, may be sufficient for numerous applications. However, secure elements can always be circumvented when all of the defenses are software-based. This means that, for new generations, there must be a physical element, either local or remote. This hardware element is a secure microprocessor known as a “secure element”. A classic example of this type of device is the smartcard, used particularly prevalently by telecom operators and banks.

Depending on whether it belongs to the world of business or public electronics, the secure element may be found in the terminal, near to it, or far away from the terminal. We shall examine the different solutions in the subsequent chapters of this book.

Virtualization also has an impact on security: the power of the Cloud, with specialized virtual machines, means that attackers have remarkable striking force at their disposal. In the last few years, hackers' ability to break encryption algorithms has increased by a factor of 5-6.

Another important point which absolutely must be integrated in networks is "intelligence". So-called "intelligent networks" have had their day, but the intelligence in this case was not really what we mean by "intelligence" in this field. Rather, it was a set of automatic mechanisms, employed to deal with problems perfectly determined in advance, such as a signaling protocol for providing additional features in the telephone system. Here, intelligence pertains to learning mechanisms and intelligent decisions based on the network status and user requests. The network needs to become an intelligent system, capable of making decisions on its own. One solution to help move in this direction was introduced by IBM in the early 2000s: "autonomic". "Autonomic" means autonomous and spontaneous – autonomous in the sense that every device in the network must be able to independently make decisions with knowledge of the situated view, i.e. the state of the nodes surrounding it within a certain number of hops. The solutions that have been put forward to increase the smartness of the networks are influenced by Cloud technology. We shall discuss them in detail in the chapter on "smart edges" (Chapter 3).

Finally, one last point, which could be viewed as the fourth revolution, is concretization – i.e. the opposite of virtualization. Indeed, the problem with virtualization is a significant reduction in performance, stemming from the replacement of hardware with software. There are a variety of solutions that have been put forward to regain the performance: software accelerators and, in particular, the replacement of software with hardware, in the step of concretization. The software is replaced by reconfigurable hardware, which can transform depending on the software needing to be executed. This approach is likely to create morphware networks, which will be described in Chapter 8.

## I.4. Conclusion

In conclusion, the world of networks is changing greatly, for the reasons listed above. It is changing more quickly than might have been expected a few years ago. One initial proposition was put forward, but failed: starting again from scratch. This is known as the "Clean Slate Approach": eliminating everything and starting again from nothing. Unfortunately, no concrete proposition has been adopted, and the transfer of IP packets continues to be the solution for data transport. However, in the numerous propositions, virtualization and the Cloud are the two main avenues which are widely used today and upon which this book focuses.

# 1

# Virtualization

In this chapter, we introduce virtualization, which is at the root of the revolution in the networking world, as it involves constructing software networks to replace hardware networks.

Figure 1.1 illustrates the process of virtualization. We simply need to write a code which performs exactly the same function as the hardware component. With only a few exceptions, which we shall explore later on, all hardware machines can be transformed into software machines. The basic problem associated with virtualization is the significant reduction in performance. On average (though the reality is extremely diverse), virtualization reduces performance by a factor of 1000: that is, the resulting software, executed on the physical machine that has been virtualized, runs 1000 times more slowly. In order to recover from this loss of performance, we simply need to run the program on a machine that is 1000 times more powerful. This power is to be found in the datacenters hosted in Cloud environments that are under development in all corners of the globe.

It is not possible to virtualize a certain number of elements, such as an antenna or a sensor, since there is no piece of software capable of picking up electromagnetic signals or detecting temperature. Thus, we still need to keep hardware elements such as the metal wires and optical links, or the transmission/reception ports of a router and a switch. Nevertheless, all of the signal-processing operations can be

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

virtualized perfectly well. Increasingly, we find virtualization in wireless systems.

More and more, to speed up the software processing, it is possible to move to a mode of concretization, i.e. the reverse of virtualization, but with one very significant difference: the hardware behaves like software. It is possible to replace the software, which is typically executed on a general machine, with a machine that can be reconfigured almost instantly, and thus behaves like a software program. The components used are derived from FPGAs (field-programmable gate arrays) and, more generally, reconfigurable microprocessors. A great deal of progress still needs to be made in order to obtain extremely fast concretizations, but this is only a question of a few years.

The virtualization of networking equipment means we can replace the hardware routers with software routers, and do the same for any other piece of hardware that could be made into software, such as switches, LSRs (Label Switching Routers), firewalls, diverse and varied boxes, DPI (Deep Packet Inspection), SIP servers, IP-PBXs, etc. These new machines are superior in a number of ways. To begin with, one advantage is their flexibility. Let us look at the example given in Figure 1.1, where three hardware routers have been integrated in software form on a single server. The size of the three virtual routers can change depending on their workload. The router uses little resources at night-time when there is little traffic, and very large at peak times in order to be able to handle all the traffic.

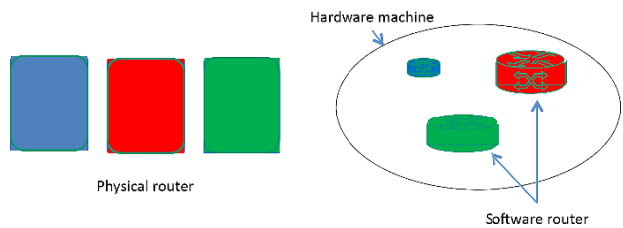



**Figure 1.1.** *Virtualization of three routers. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

Energy consumption is another argument in favor of virtualization. Whilst, to begin with, consumption would rise because we are adding an extra piece of software (the hypervisor), it is possible to share the resources more effectively, and move those resources, grouping them together on physical machines, and put other machines, which have become idle, on standby.

A physical machine can accommodate virtual machines if, as mentioned above, we add a hypervisor, which is a software program that enables multiple virtual machines to run simultaneously. In actual fact, they only appear to run simultaneously at a macroscopic scale. Examined more closely, at the microscopic scale, we see that the virtual machines are executed sequentially one after the other. In the context of virtual servers, this serial execution is not a problem. In the area of networks, it may become a problem for real-time applications, which require a very short response time. Each virtual machine's processing time must be sufficiently short to give the impression that all the virtual machines are being executed in parallel. Figure 1.2 illustrates the architecture of virtualization.

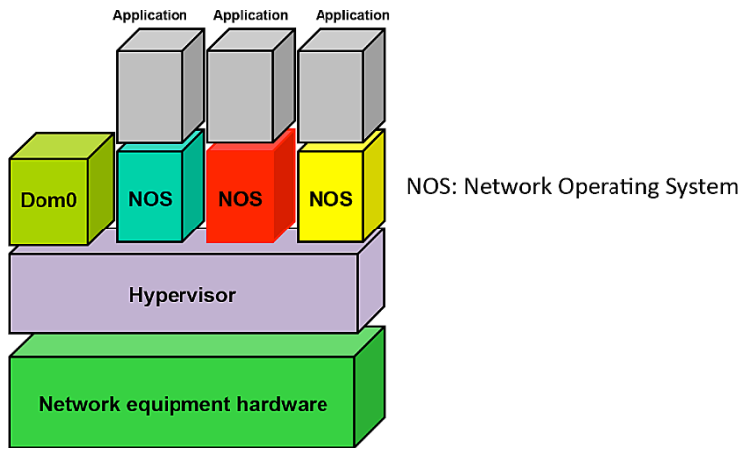



**Figure 1.2.** *A virtualized machine. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

The hypervisor is a virtual machine monitor (VMM), which is often open source. Hypervisors operate on standard hardware platforms. In addition to the VMM, running directly on the physical hardware, the architecture generally comprises a number of domains running simultaneously, as we have seen, on top of the hypervisor, called virtual machines. Each virtual machine may have its own operating system and applications. The VMM controls access to the hardware from the various domains, and manages the sharing of the resources between the different domains. Thus, one of the VMM's main tasks is to isolate the different virtual machines, so that the execution of one virtual machine does not affect the performances of the others.

All peripheral drivers are kept in an isolated domain specific to them. Known as "domain zero" (dom0), it offers a reliable and effective physical support. Dom0 has special privileges in comparison to other domains, known as "user domains" (domU) and, for example, has unfettered access to the hardware of the physical machine. User domains have virtual drivers, and operate as though they have direct access to the hardware. However, in reality, those virtual driver communicate with the dom0 in order to access the physical hardware.

The hypervisor virtualizes a single physical network interface, de-multiplexing the incoming packets from the physical interface to the user domains and, conversely, multiplexing the outgoing packets generated by those user domains. In this procedure, known as virtualization of the network input/output, the domain 0 directly accesses the input/output peripherals, using their native drivers, and performs input/output operations on behalf of the domUs.

The user domains employ virtual input/output peripherals, controlled by virtual drivers, to ask the dom0 for access to the peripheral. Each user domain has its own virtual network interfaces, known as foreground interfaces, which are required for network communications. The background interfaces are created in the dom0, corresponding to each foreground interface in a user domain, and act as proxy for the virtual interfaces in the dom0. The foreground and background interfaces are connected to one another via an input/output channel, which uses a zero-copy mechanism to match the physical page containing the packet and the target domain. Thus, the

packets are exchanged between the background and foreground interfaces. The foreground interfaces are perceived by the operating systems, working on the user domains, as real interfaces. However, the background interfaces in the dom0 are connected to the physical interface and to one another via a virtual network bridge. It is the default architecture, called "bridge mode", used for instance by the Xen hypervisor, which was certainly one of the first to appear. Thus, both the input/output channel and the network bridge establish a path for communication between the virtual interfaces created in the user domains and the physical interface.

## 1.1. Software networks

Virtual machines, in turn, can be used to create virtual networks, which are also known as software networks. For this purpose, we need to link virtual machines together in the same way as we would connect different physical machines. Of course, the communication links must be shared between the different software networks. A set of software networks is represented in Figure 1.3.

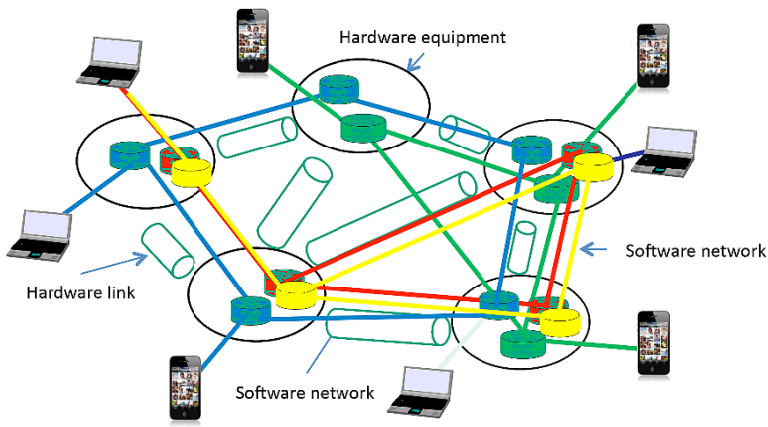



**Figure 1.3.** *A set of software networks. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

Each software network may have its own architecture and its own characteristics. One software network could be devoted to a VoIP service, another to IP-TV, a third to a highly-secure application, a fourth to channeling professional applications, a fifth for asynchronous applications such as electronic messaging, etc. We could, in fact, practically create a software network for each user. The personalized software network is set up at the moment when the user connects. It is eliminated when the user signs out. However, this solution does not scale up, and today we are limited to a number of software networks suited to the hardware capacity of the underlying physical infrastructure. Each software network receives resources allocated to it on the basis of the user demands.

It should be noted that, in general, the virtual nodes are found in datacenters, which may be of varying size and importance: enormous central datacenters, regional datacenters, local datacenters and small datacenters such as femto-datacenters. We shall come back later on to the choices which may be made in this field.

One of the characteristics of software networks is that the virtual machines can be migrated from one physical machine to another. This migration may be automated based on whether a node is overloaded or out of order.

In the physical nodes which support the software networks, we can add other types of virtual machines such as firewalls, SIP servers for VoIP, ADSL router, etc. The networks themselves, as stated above, may obey a variety of different protocol architectures such as TCP/IPv4, UDP/IPv4, IPv6, MPLS, Ethernet Carrier Grade, TRILL, LISP, etc.

Isolation is, of course, a crucial property, because it is essential to prevent a problem on one software network from having repercussions for the other networks. The handover of streams from one software network to another must take place via a secure gateway outside of the data plane. This is absolutely necessary to prevent contamination between networks, such as a complete shutdown for a network attacked by a distributed denial of service (DDOS).

## 1.2. Hypervisors

Clearly, virtualization needs hardware, which can be standard. We speak of commodity hardware, with open specifications, produced *en masse* to achieve particularly low prices. There are various ways of placing virtual machines on physical equipment, and they can be classified into three broad categories, as shown in Figures 1.4 to 1.6.

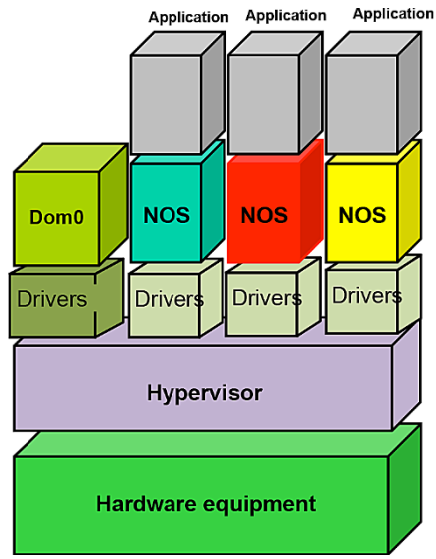



**Figure 1.4.** *Para-virtualization. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

A para-virtualization hypervisor, also called "bare metal", is a program which is executed directly on a hardware platform. This platform is able to support guest operating systems with their drivers. The classic hypervisors in this category include Citrix Xen Server (open source), VMware vSphere, VMware ESX, Microsoft Hyper-V Server, Bare Metal and KVM (open source). These programs are also known as type-1 hypervisors.

The second category of hypervisor, or type 2, is a program which is executed within a different operating system. A guest operating

system is executed above the hardware and requires an emulator to be executed on the host operating system. The guest operating systems are unaware that they are virtualized, so they do not require any modifications. Examples of this type of virtualization would include Microsoft Virtual PC, Microsoft Virtual Server, Parallels Desktop, Parallels Server, Oracle VM Virtual Box (free), VMware Fusion, VMware Player, VMware Server, VMware Workstation and QEMU (open source).

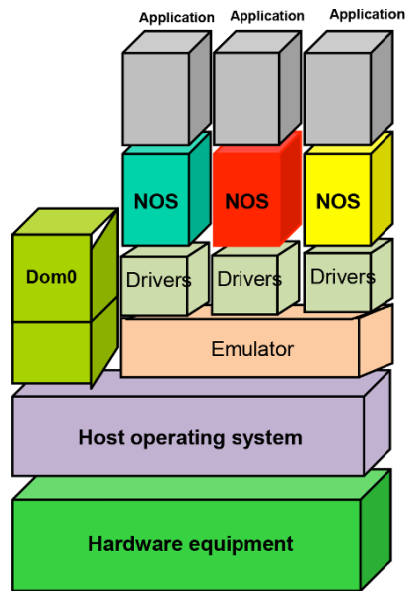



**Figure 1.5.** *Virtualization by emulation. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

The third type leaves behind the previous hypervisor systems, running several machines simultaneously as shown in Figure 1.6. We tend to speak of an isolator. An isolator is a program which isolated the execution of the applications in an environment, called the context, or indeed the zones of execution. Thus, the isolator is able to run the same application multiple times in a multi-instance mode. This solution performs very well, because it does not cause any overload, but the environments are more difficult to isolate.

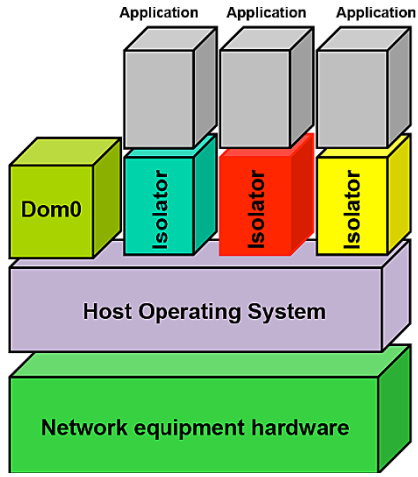



**Figure 1.6.** *Virtualization by execution zones. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

In summary, this last solution facilitates the execution of the applications in execution zones. In this category, we can cite the examples of Linux-Vserver, chroot, BSD Jail and Open VZ.

Software networks have numerous properties which are novel in comparison to hardware networks. To begin with, we can easily move virtual machines around, because they are simply programs. Thus, we can migrate a router from one physical node to another. Migration may occur when a physical node begins to fail, or when a node is overloaded, or for any other reason decided upon in advance. Migration of a node does not actually involve transporting the whole of the code for the machine, which would, in certain cases, be rather cumbersome and time-consuming. In general, the program needing to be transported is already present in the remote node, but it is idle. Therefore, we merely need to begin running the program and send it the configuration of the node to be moved. This requires the transmission of relatively little data, so the latency before the migrated machine starts up is short. In general, we can even let both machines run at once, and change the routing so that the data only flow through the migrated node, and we can then shut down the first router.

More generally, we carry out what is known as urbanization: we migrate the virtual machines to different physical machines until we obtain optimal performance. Urbanization is greatly used for optimization in terms of energy consumption or workload distribution, but also to optimize the cost of the software networks or to make the network highly reliable or resilient. For example, in order to optimize energy consumption, we need to bring together the virtual machines on shared nodes and switch off all the nodes which are no longer active. In actual fact, these machines would not be shut down but rather placed on standby, which does still consume a small amount of energy, but only a very small amount. The major difficulty with urbanization arises when it is necessary to optimize all operational criteria, because they are often incompatible – e.g. optimizing consumption and performance at the same time.

A very important characteristic is isolation: the software networks must be isolated from one another, so that an attack on one network does not affect the other networks. Isolation is complex, because simultaneously, we need to share the common resources and be sure that, at all times, each network has access to its own resources, negotiated at the time of establishment of the software network. In general, a token-based algorithm is used. Every virtual device on every software network receives tokens according to the resources attributed to it. For example, for a physical node, ten tokens might be distributed to network 1, five tokens to network 2 and one token to network 3. The networks spend their tokens on the basis of certain tasks performed, such as the transmission of $n$ bytes. At all times, each device can have its own tokens and thus have a minimum data rate, determined when the resources were allocated. However, a problem arises if a network does not have packets to send, because then it does not spend its tokens. A network may have all of its tokens when the other networks have already spent all of theirs. In this case, so as not to immobilize the system, we allocate negative tokens to the other two networks, which can then surpass the usage rate defined when their resources were allocated. When the sum of the remaining tokens less the negative tokens is equal to zero, then the machine's basic tokens are redistributed. This enables us to maintain isolation whilst still sharing the hardware resources. In addition, we can attach a certain

priority to a software network whilst preserving the isolation, by allowing that particular network to spend its tokens as a matter of priority over the other networks. This is relative priority, because each network can, at any moment, recoup its basic resources. However, the priority can be accentuated by distributing any excess resources to the priority networks, which will then always have a token available to handle a packet. Of course, isolation requires other characteristics of the hypervisors and the virtualization techniques, which we shall not discuss in this book.

Virtualization needs to be linked to other features in order to fully make sense. SDN (Software-Defined Networking) is one of the paradigms strongly linked to virtualization, because it involves the uncoupling of the physical part from the control part. The control part can be virtualized and deported onto another machine, which enables us, for example, to have both a far great processing power than on the original machine, and also a much larger memory available.

## 1.3. Virtual devices

All devices can be virtualized, with the exception of those which handle the reception of terrestrial and wireless signals, such as electromagnetic signals or atmospheric pressure. For example, an antenna or thermometer could not be replaced by a piece of software. However, the signal received by that antenna or thermometer can be processed by a virtual machine. A sensor picking up a signal can select an appropriate virtual machine to process the signal in order to achieve a result that is appropriate for the demand. The same antenna might, for example, receive signals from a Wi-Fi terminal but also signals from a 4G terminal. On the basis of the type of signal, an initial virtual machine determines which technology is being used, and sends the signal to the virtual machine needed for its processing. This is known as SDR (Software-Defined Radio), which is becoming increasingly widely used, and enables us to delocalize the processing operation to a datacenter.

The networking machines which we know can always be virtualized, either completely or at least partially: the processing part,

the control part and the management part. Thus, today, we can uncouple a physical machine which, in the past, was unique, into several different machines – one of them physical (e.g. a transceiver broadcasting along a metal cable) and the others virtual. One of the advantages of this uncoupling is that we can deport the virtual parts onto other physical machines for execution. This means that we can adapt the power of the resources to the results we wish to obtain. Operations originating on different physical machines can be multiplexed onto the same software machine on a single physical server. This solution helps us to economize on the overall cost of the system, but also on the energy expended, by grouping together the necessary power using a single machine that is much more powerful and more economical.

Today, all legacy machines in the world of networking have either been virtualized already or are in the process of being virtualized – Nodes-B for processing the signals from 3G, 4G and soon 5G mobile networks, HLRs and VLRs, routers, switches, different types of routers/switches such as those of MPLS, firewalls, authentication or identity-management servers, etc. In addition, these virtual machines can be partitioned so they execute on several physical machines in parallel.

We can appreciate the importance of the Cloud and associated datacenters, because they are placed where the processing power is available at a relatively low cost, as is the memory space needed to store the virtual machines and a whole range of information pertaining to the networks, clients and processing algorithms. The tendency with server virtualization is to focus on huge datacenters, but with the help of distribution, we are seeing smaller and smaller datacenters.

## 1.4. Conclusion

Virtualization is the fundamental property of the new generation of networks, where we make the move from hardware to software. Whilst there is a noticeable reduction in performance at the start, it is compensated by more powerful, less costly physical machines. Nonetheless, the opposite move to virtualization is crucial: that of

concretization, i.e. enabling the software to be executed on reconfigurable machines so that the properties of the software are retained and top-of-the-range performances can again be achieved.

Software networks form the backbone of the new means of data transport. They are agile, simple to implement and not costly. They can be modified or changed at will. Virtualization also enables us to uncouple functions and to use shared machines to host algorithms, which offers substantial savings in terms of resources and of qualified personnel.

# 2

# SDN (Software-Defined Networking)

The SDN (Software-Defined Networking) technology is at the heart of this book. It was introduced with virtualization, enabling networking devices to be transformed into software. Associated with this definition, a new architecture has been defined: it decouples the data level from the control level. Up until now, forwarding tables have been computed in a distributed manner by each router or switch. In the new architecture, the computations for optimal control are performed by a different device, called the controller. Generally, the controller is centralized, but it can perfectly well be distributed. Before taking a closer look at this new architecture, let us examine the reasons for this new paradigm.

The limitations of traditional architectures are becoming significant: at present, modern networks no longer optimize the costs at all (i.e. the CAPEX and OPEX). In addition, the networks are not agile. The time to market is much too long, and the provisioning techniques are not fast enough. In addition, the networks are completely unconnected to the services, and the following points need to be taken into account in the new SDN paradigm:

– overall needs analysis;

– dynamic, rather than static, configuration;

– dynamic, rather than static, policies used;

– much greater information feedback than is the case at present;

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

– precise knowledge of the client and of his/her applications, and more generally his/her requirements.

## 2.1.The objective

The objective of SDN (Software-Defined Networking) is to reduce costs by virtualization, automation and simplification. For this purpose, SDN facilitates the customization of the networks, a very short set-up time and a network deployment with the right quality of service rather than a general quality of service.

The architecture of SDN can be summarized with three fundamental principles, as shown in Figure 2.1. The first is the decoupling of the physical and virtual layers (hardware and software). This enables virtual devices to be loaded on hardware machines. The network becomes independent of the hardware. This is not completely realized in reality, because there may be a dependence on the hypervisor, and certain device manufacturers exploit this point to force customers to buy their own hardware. The second principle pertains to the devices connected to the network, which must perceive no difference between a hardware or software machine. This new environment enables us to change the network without having to do anything to the host machines. Finally, the third principle is that of automation – the best possible automation – of the operations carried out on the network, whether for management or for control.

The new environments are defined by three domains which characterize an information- and operation system for a company: storage, processing and the network. However, in order for the environment to be able to be executed without problems, we must add the domains of security and management & control. Today, all five of these domains need to be in place to constitute a company's complete information and operation system. These five domains are shown in Figure 2.2.

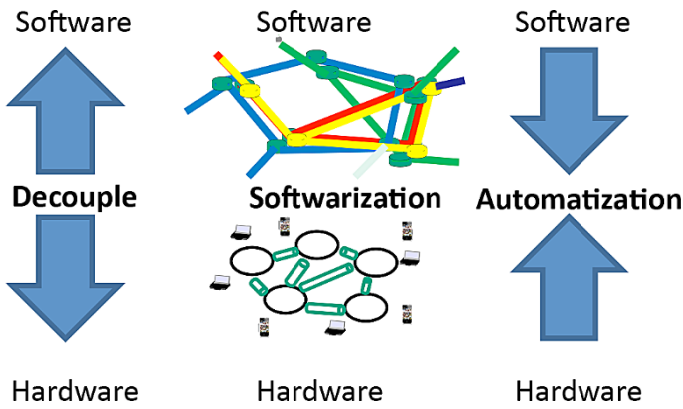



**Figure 2.1.** *The three basic principles. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

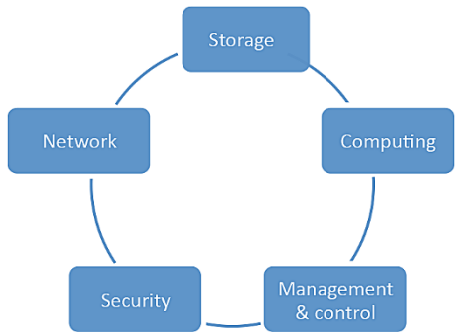



**Figure 2.2.** *The five domains necessary for the life of a company*

The five domains described above can be put in place by way of virtual machines associated with each of the domains. The informational and operational environment can thus be concentrated in the Cloud in the form of virtual machines distributed in datacenters. This environment is illustrated in Figure 2.3 by datacenters containing the virtual machines necessary for the construction of the whole.

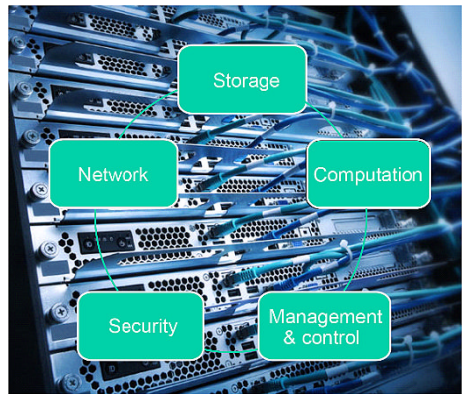



**Figure 2.3.** *Virtualization of the five domains*

In addition to this environment, there are applications which may be of two types: business applications and applications to control or orchestrate the environment itself. The search for new products has therefore turned toward autopilot systems, which are also referred to as orchestrators in the literature. The complete environment is illustrated in Figure 2.4, which shows the importance of the orchestrator in the general architecture of company informational and operational systems.

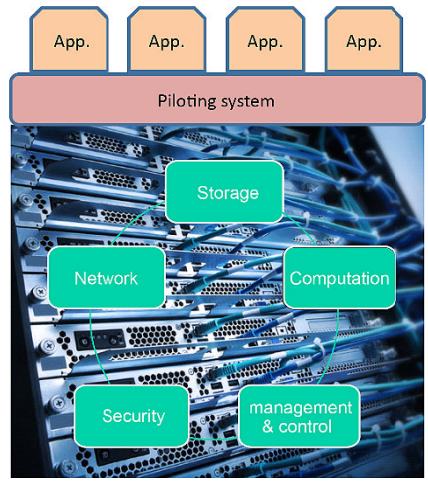



**Figure 2.4.***The pilot program*

## 2.2. The ONF architecture

In order for this new world of SDN to have a chance of being successful, it has to be standardized. This standardization was carried out by the ONF (Open Network Foundation), which was set up under the auspices of large companies in California, following the proposal of this architecture by Stanford University and Nicira.

The architecture proposed by the ONF is shown in Figure 2.5. It comprises three layers. The bottom layer is an abstraction layer, which decouples the hardware from the software, and is responsible for data transport. This level describes the protocols and algorithms which enable IP packets to advance through the network to their destination. This is called the infrastructure plane. The second layer is the control plane. This plane contains the controllers providing control data to the data plane so that the data are channeled as effectively as possible. The ONF's vision is to centralize control in order to facilitate the recovery of a great deal of information on all the clients. The centralized controller enables obtaining a sort of intelligence. The infrastructure to be managed is distributed between the controllers. Of course, we need to take account of the problems of a centralized environment, and therefore duplicate the decision elements.

Controllers carry out different functions, such as the provision of infrastructure, the distribution (or otherwise) of loads on different network devices to optimize performances or reduce energy consumption. The controller is also in charge of the deployment of firewalls and servers necessary for the proper operation of the network and a management system. These different machines must be put in the most appropriate places.

Finally, the uppermost layer, the application plane, is responsible for the applications needed by the clients and for their requirements in terms of networking, storage, computation, security and management. This layer introduces the programmability of the applications, and sends the controller all of the necessary elements to open the software networks tailored to the needs of the applications. This layer also

includes a few very special applications such as the orchestrator, who sends the controller the necessary information to open up the network which corresponds to the application. Any new service can be introduced quickly, and will give rise to a specific network if it cannot be embedded on a pre-existing network.

The ONF architecture is shown in Figure 2.5, with its three layers: the application layer and programmability, the control layer with centralized intelligence, and abstraction at the infrastructure layer. We shall come back to look at the interfaces between these layers, which are important for the compatibility of products from different vendors. ONF has standardized the intermediary layer and the interfaces. Certain parts are taken up by other standardization organizations so as to conform to the legal standards.

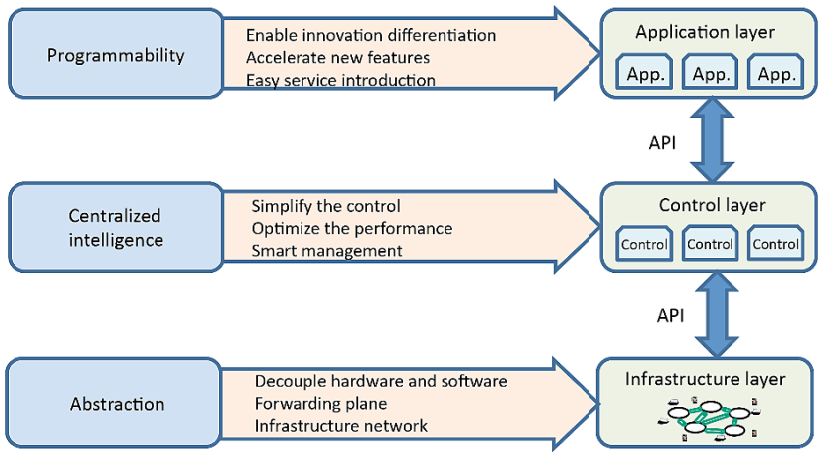



**Figure 2.5.** *The ONF architecture*

The ONF's general architecture can actually be more detailed, as is illustrated by Figure 2.6. Once again we see the infrastructure layer, but it is expanded into two planes: the physical plane and the logical plane. The physical plane is in charge of all the hardware, and more generally, the physical infrastructure. The logical plane, for its part,

corresponds to the establishment of the software networks constructed on the basis of virtual machines, sharing the physical infrastructure in accordance with the rules deriving from the higher layers. This vision of the architecture enables us to clearly discern the hardware and the network which exist in companies from the software, which is added to offer the necessary flexibility. It is clear that this new architecture requires more hardware, and therefore the idea is to add datacenters ranging in size from very small to very large, depending on the size of the company. Telecoms operators have not missed this opportunity, and have entered into the market as Cloud providers. Companies such as Amazon and Google have gone directly for the goal, putting in place the infrastructure necessary to become major players in the world of telecommunications.

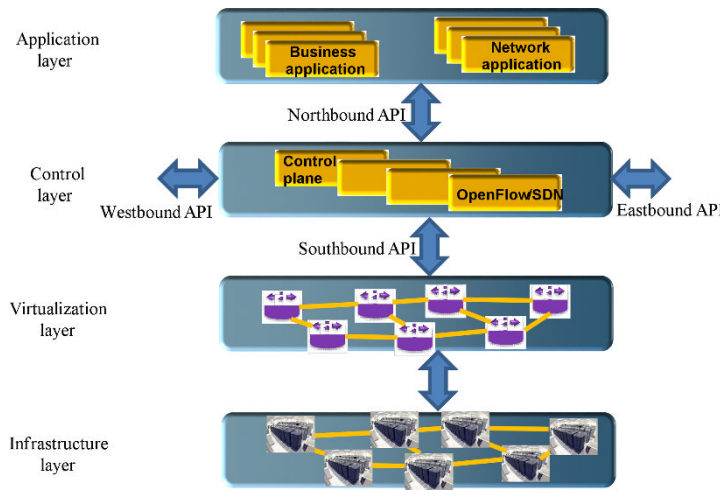



**Figure 2.6.** *The SDN architecture*

In the architecture shown in Figure 2.6, we see the control layer and the application layer with the northbound and southbound APIs (Application Programming Interfaces) between those layers, and the eastbound and westbound APIs with other controllers. The northbound interface facilitates communication between the application level and the controller. Its purpose is to describe the needs of the application and to pass along the commands to

orchestrate the network. Later on, we shall describe the current standards governing this interface. The southbound interface describes the signaling necessary between the control plane and the virtualization layer. With this aim in mind, the controller must be able to determine the elements that will make up the software network for which it is responsible. In the other direction, the current network resource consumption must be fed back so that the controller has as full a view as possible of the usage of the resources. The bandwidth necessary for the feeding back of these statistics may represent a few percent of the network's capacity, but this is crucial for optimization which will improve performance by much more than a few percent.

In addition to the two interfaces described above, there are also the eastbound and westbound interfaces. The eastbound interface enables two controllers of the same type to communicate with one another and make decisions together. The westbound interface must also facilitate communication between two controllers, but ones which belong to different sub-networks. The two controllers may be compatible but they may also be incompatible and in this case, a signaling gateway is needed.

Figure 2.7 shows a number of important open-source programs that have been developed to handle a layer or an interface. Starting from the bottom, in the virtualization layer, network virtual machines are in the process of standardization by the ETSI in a working group called NFV (Network Functions Virtualization), which we shall revisit in detail later on. Here, let us simply note that the aim of NFV is to standardize network functions with a view to virtualizing them and facilitating their execution in different places from the original physical machine.

The control plane includes the controllers. One of the best known is OpenDaylight – an open-source controller developed collaboratively by numerous companies. This controller, as we shall see later on, contains a large number of modules, often developed in the interest of the particular company that carried out that work. Today, OpenDaylight is one of the major pieces in the CISCO architecture, but also that of other manufacturers. Later on we shall

detail most of the functions of OpenDaylight. Of course, there are many other controllers – particularly Open Source ones – such as OpenContrail, Flood Light, etc.

The uppermost layer represents the Cloud management systems. It is roughly equivalent to the operating system on a computer. It includes Open Stack, which was the system which was most favored by the developers, but again, many other products exist, both open source and proprietary.

The southbound interface is often known by the term OpenFlow. OpenFlow is a signaling system between the infrastructure and the controller. This protocol was designed by Nicira and has led to a *de facto* standard from the ONF. OpenFlow transports information that properly defines the stream in question to open, modify or close the associated path. OpenFlow also determines the actions to be executed in one direction or the other over the interface. Finally, OpenFlow facilitates the feeding back of measurement information performed over the different communication ports, so that the controller has a very precise view of the network.

The northbound and southbound interfaces have been standardized by the ONF, to facilitate compatibility between Cloud providers, the control software and the physical and virtual infrastructure. Most manufacturers conform to these standards to a greater or lesser degree, depending on the interests in the range of hardware already operating. Indeed, one of the objectives is to allow companies which have an extensive range to be able to upgrade to the next generation of SDN without having to purchase all new infrastructure. A transitional period is needed, during which we may see one of two scenarios:

– the company adapts the environment of the SDN to its existing infrastructure. This is possible because the software layer is normally independent on the physical layer. The company's machines must receive one of the hypervisors compatible with the manufacturer's products. However, it is important to add to or update the infrastructure so that it gains at least 10%, but preferably 20 or 30%, to be able to handle numerous logical networks;

– the company implements the new SDN architecture on a new part of its network, and increases it little by little. This solution means that both the old generation of the network and the new need to be capable of handling the demand.

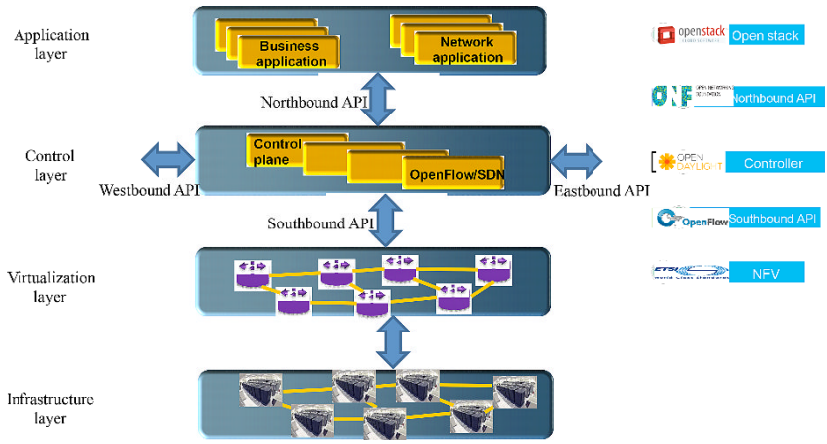



**Figure 2.7.** *Example of Open Source developments*

Now let us go into detail about the different layers. Once again, we shall start with the bottom layer. The physical layer is quintessential and is designed around the most generic possible hardware devices in order to obtain the best cost/benefit ratio, but these devices are still suited for networking i.e. with the necessary communications cards and the appropriate capacity to host the intended software networks. It is clear that performance is of crucial importance, and that the physical infrastructure has to be able to cope with what is at stake. One of the priorities, though, is to minimize the physical layer to avoid consuming too much in the way of energy resources. With this goal in mind, the best thing to do is to have an algorithm to appropriately place the virtual machines, with a view to putting the highest possible number of physical machines on standby outside of peak hours. Urbanization is becoming a buzz word in these next-generation networks. Unfortunately, urbanization algorithms are still in the very early stages of development, and are not capable of dealing

with multicriteria objectives: only one criterion is taken into account – e.g. performances, or energy expenditure. Energy optimization asks for an urbanization definitely different of load sharing – i.e. channeling the data streams along common paths in order to be able to place a maximum number of physical machines on standby. The difficulty in the latter case is being able to turn the resources back on as the workload of the software networks increases again. Certain machines, such as virtual Wi-Fi access points, are difficult to wake from standby mode when external devices swish to connect wirelessly. We need electromagnetic sensors that are capable of detecting these mobile terminals and sending them an Ethernet frame at the Wi-Fi access point with the function Wake-on-LAN.

## 2.3. NFV (Network Functions Virtualization)

The purpose of NFV (Network Functions Virtualization) is to decouple the network functions from the network equipment. This decoupling enables us to position the software performing the functions of a device on a different machine than the device itself. This means we can place the operational programs of a machine in a datacenter within a Cloud. Standardization is being done by a working group from the ETSI, which is a European standardization body, but in this particular case, the project is open to all operators and device manufacturers from across the world. Over two-hundred members are taking part in this standardization effort. The objective of this initiative is to define the architecture to virtualize the functions included in the networking devices, and to clearly define the challenges needing to be overcome. The standardization tasks are being carried out by five separate working groups, described in detail below.

The first group, “Architecture of the Virtualization”, has the objective of producing a reference architecture for a virtualized infrastructure and points of reference to interconnect the different components of that architecture. The second group, “Management and

Orchestration", is charged with defining the rollout, instantiation, configuration and management of network services that use the NFV infrastructure. The third group, "Software Architecture", aims to define the reference software architecture for the execution of virtualized functions. The fourth group, "Security Expert Group", as the name suggests, works on the security of the software architecture. Finally, the last group, "Performance and Portability Expert Group", needs to provide solutions to optimize performances and manage the portability of the virtual machines representing the functions of this new environment.

Figure 2.8 shows a number of appliances handled by NFV, such as firewalls or SBCs (Session Border Controllers). These functions are moved to a powerful physical machine or to a virtual machine in a datacenter.

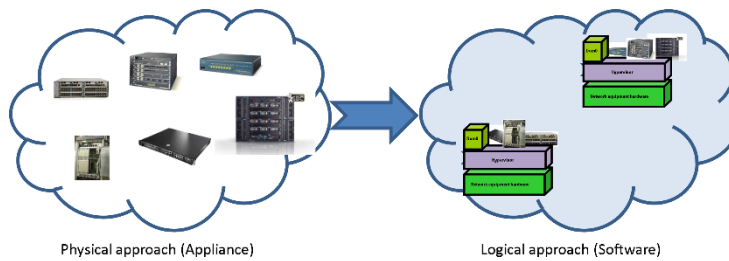



**Figure 2.8.** *NFV (Network Functions Virtualization)*

Figure 2.9 goes into a little more detail as to the machines whose functions are externalized. ETSI's aim is to standardize the virtual machines used to perform these functions. The power of the server determines the rate at which the functions are executed. This makes sense for the functions of a firewall or deep packet inspection (DPI), which require extremely high processing power to determine the threats or applications passing through the network in real time.

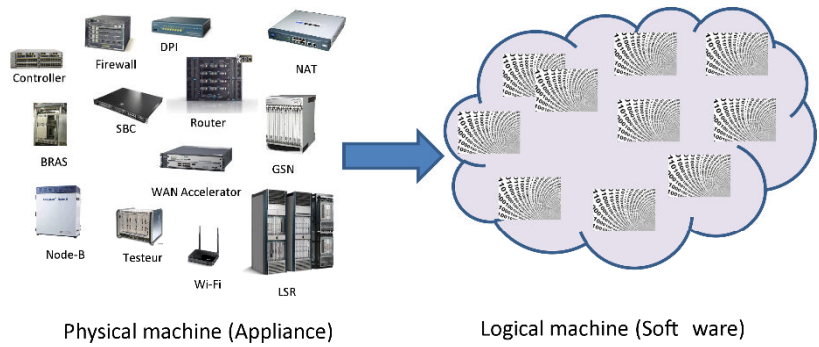



**Figure 2.9.** *NFV machines*

## 2.4. OPNFV

OPNFC, or Open NFV, is a new movement which is a collaborative project from the Linux Foundation. The objective is to create a platform to speed up the rise of NFV. Another objective of OPNFV is to increase the agility of services by facilitating better usage of the underlying functions. Telecom operators are particularly interested by this initiative; they hope to obtain a referential Open Source platform to validate the interoperability of NFVs in a multi-provider context. OPNFV is also an excellent opportunity for the creation of an open platform that could be used by all participants. OPNFV will also encourage cooperation between manufacturer, with the aim of driving NFV forward and ensuring consistency, performance and interoperability between virtualized network infrastructures. OPNFV is in close cooperation with the NFV program at ETSI, amongst other standardizing bodies, to ensure a coherent implementation of the NFV standard. The first thing that is expected of OPNFV is to provide an NFV Infrastructure (NFVI), Virtualized Infrastructure Management (VIM), and APIs to other NFV elements. This collection forms the basic infrastructure necessary for Management and Network Orchestration (MANO). The standards are being drawn up in collaboration with the main Open-Source projects. OPNFV is working with a large number of these projects to coordinate the integration of NFV.

## 2.5. Southbound interface

The southbound interface is situated between the controller and the devices on the virtualization plane. This signaling protocol passes the configuration commands in one direction and the statistical information feedback in the other. There are many open source proposals for some under development. An initial list of protocols for this southbound interface is:

– OpenFlow from the ONF;

– I2RS (Interface to Routing Systems) from the IETF;

– Open vSwitch Data Base (OvSDB);

– Net Conf;

– SNMP;

– LISP;

– BGP.

We shall detail the most significant protocols in the next chapter. However, to introduce this southbound interface, let us look at the most iconic protocol: OpenFlow. This signaling is illustrated in Figure 2.10. It takes place between the controller and the network devices. The data transported obey the trilogy "match-action-statistics"– in other words, we need to:

– determine the streams uniquely by matching a certain number of elements such as addresses or port numbers;

– specify the actions transmitted from the controller to the networking devices, such as the rows in a forwarding table or switch table, or more generally a forwarding device;

– transfer tables containing statistics on the degree of use of the ports, transmission lines and nodes, such as the number of bytes sent through a given port.

This solution has been standardized by the ONF (Open Network Foundation), and we shall describe it in greater detail in the next chapter.

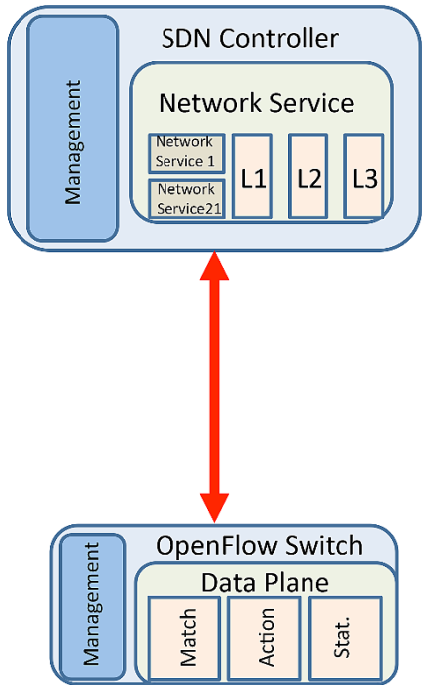



**Figure 2.10.** *The signaling protocol OpenFlow*

## 2.6. The controller

The controller, as its name indicates, is designed to control the data plane and received from the application layer the necessary elements to determine the type of control that needs to be exerted. This position is illustrated in Figure 2.11.Thus, the controller must receive policy rules to be respected and, on the basis of the description of the applications to be taken into account, it deduces the actions needed on the networking equipment. The actions can be carried out on routers, switches, firewalls, load balancers, virtual VPNs and other hardware.

A very great many open source controllers have been developed. OpenDaylight represents a good example; this open source software was developed by the Linux foundation. A good forty companies devoted experienced developers to the project. The controller as such has numerous decision-making modules and very numerous

northbound and southbound interfaces. We shall describe the latest release of OpenDaylight hereinafter.

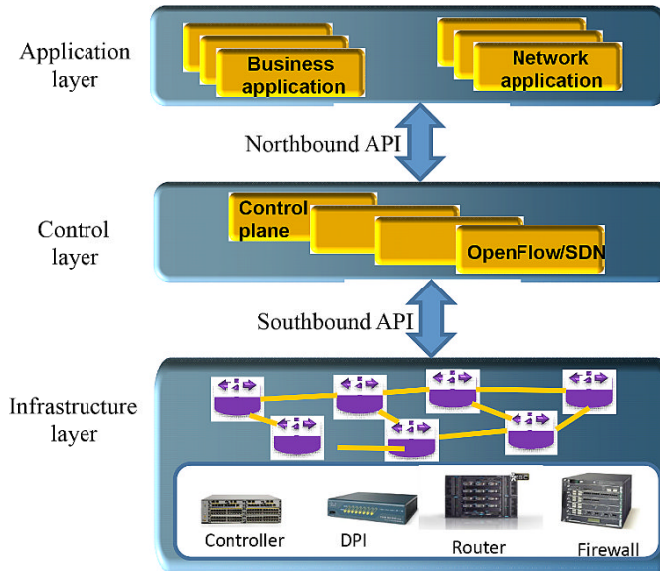



**Figure 2.11.** *The control layer and its interfaces*

The controller contains modules that handle different functions necessary for the proper operation of the network. Of these functions, one of the most important is that of a "load balancer". In fact, this term denotes algorithms which determine the best paths to follow on the data plane. This module must decide, on the basis of the statistics received, which nodes in the network infrastructure the packets should be sent through. This decision should optimize the user demands or, more specifically, the user applications. Load balancing is essentially valid during peak hours. During other times, the load balancer must determine the most appropriate paths, on the basis of the users' requirements. Load balancing becomes load unbalancing: unlike at peak times, we need to channel as many streams as possible through common paths so as to be able to shut down the maximum possible number of intermediary nodes. This function is represented diagrammatically in Figure 2.12.

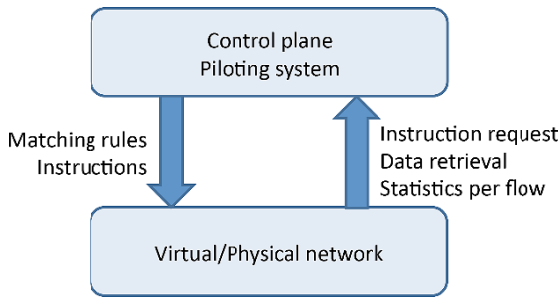



**Figure 2.12.** *The load-balancing protocol*

## 2.7. Northbound interface

The northbound interface between the application plane and controller passes information pertaining to the applications' needs so that the controller can open up the best possible software network with the appropriate qualities of service, adequate security and the necessary management for the operations to be able to be carried with no problems. The basic protocol for these transmissions is based on the REST (Representative State Transfer) API. This application interface must enable us to send the information necessary for the configuration, the operations, and the measurements. We find this protocol, based on REST ful, integrated into numerous Cloud management systems, and in particular in the interfaces of most of these systems, such as:

– Open Stack, in which the REST API is integrated;

– those of the service providers;

– the API Virtual Private Cloud (VPC).

The representations facilitated by the REST protocol enable us to pass the information over the northbound interface with the following properties:

– each resource is identified individually;

– the resources can be manipulated by representations;

– the messages are self-descriptive: they explain their nature by themselves;

– each access to the subsequent states of the application is described in the present message.

## 2.8. Application layer

The application layer is essentially formed of Clouds which host the application virtual machines. These may be business machines or network element management machines – responsible, for instance, for the managing of handovers or determination of the best access, at any given time, for a multi-technology terminal. Thus, this layer essentially contains the operating systems for the Cloud. The best-known such system is, undoubtedly, Open Stack. It is a Cloud management system which controls large sets of resource offering processing power, storage and network resources. Open Stack has already had more than 10 releases in only 4 years (April 2011 to April 2015). The latest version is Kilo, with Liberty due to arrive at the end of 2015. Open Stack is open-source software with an Apache license.

The Open Stack architecture is illustrated in Figure 2.13.It is modular, and contains numerous modules developed in parallel, such as Nova for computation, Swift for storage, Glance for the imaging service, Dashboard for the settings and control panel, etc.

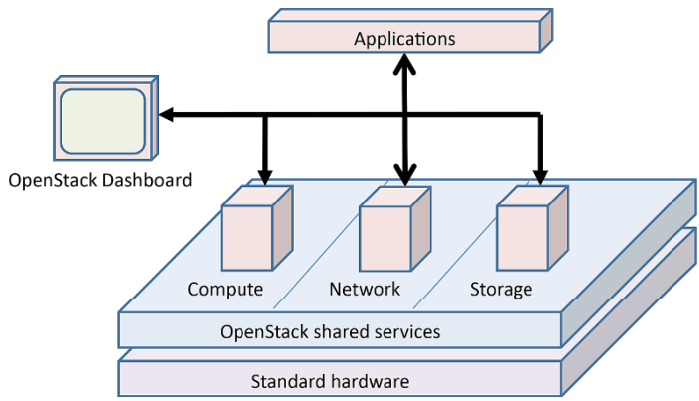



**Figure 2.13.** *The Open Stack system*

The part which is of interest to us most in this book is Neutron, which handles the network module. In this context, Open Stack provides flexible network models to take care of the applications. In particular, Open Stack Neutron manages the IP addresses, and allows for static addresses or uses DHCP. Users can create their own network in which SDN technology is used. Open Stack has numerous extensions, such as IDS (Intrusion Detection Systems), load balancing, the option to deploy firewalls and VPNs.

To conclude this section and summarize the SDN architectures in place, in Figure 2.14 we have shown the different components that have been put in place to achieve overall operation. The top and bottom parts represent the Cloud and the physical/logical networks. Between these two points, management and control of the network and of the applications needs to take place. In terms of the business application, we find sets of software modules – mostly open source – to deploy cloud-computing infrastructures, and more specifically IaaS (infrastructure as a service). On the other hand, we find the applications to establish a virtualized network structure, with the commands necessary to handle the business applications.

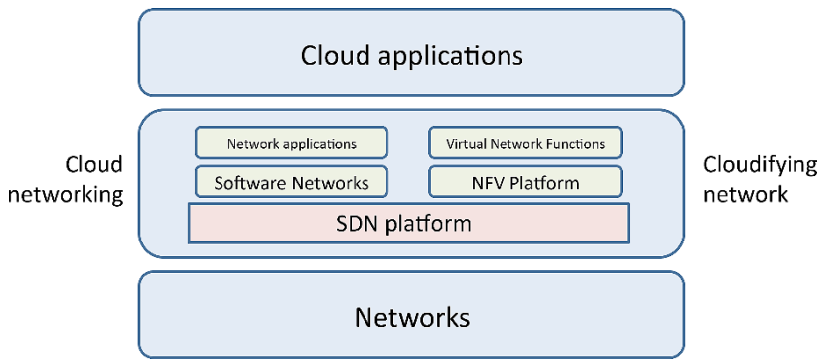



**Figure 2.14.** *The overall architecture of SDN solutions*

## 2.9. Urbanization

We have already mentioned the issue of urbanization of virtual machines in a network. Let us now look again at this concept in a little

more detail. It involves placing the virtual machines in the network, i.e. in the Cloud, so that optimum performance is attained. Whilst performance is, obviously, very important, in today's world the cost of datacenters, and therefore of networks, is based mainly on energy expenditure. To clarify the issue, Figure 2.15 shows the cost of a datacenter, distributed between the infrastructure and maintenance costs.

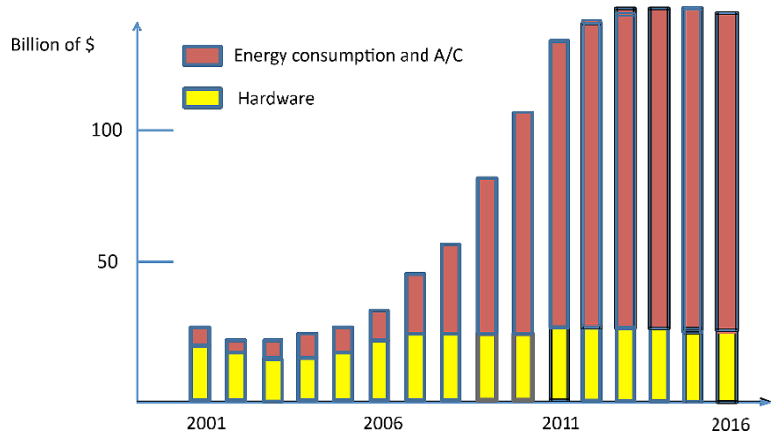



**Figure 2.15.** *The cost of a datacenter environment. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

Urbanization is an initial response to this demand to economize on power supply, attempting to bring together the virtual machines on common servers so as to be able to put a large number of servers, which have become idle, on standby. This solution is very useful at night and at low-demand times of the day. During the peak hours, the servers need to be awakened again, with a wider distribution of the virtual machines between the different servers.

Urbanization also takes account of cost factors, leading us to use the physical machines at night, and migrate the virtual machines, having them go all around the world in the space of 24 hours. Evidently, this solution is viable only for light virtual machines which can be shifted with no problems, and is absolutely not appropriate for "big data" processing.

Urbanization may also affect other criteria such as the availability of the network, by facilitating access to emergency paths and virtual machines, distributed appropriately so that there are multiple access paths. Reliability also pertains to the same sensitive points, and solutions to reliability issues may be found through virtualization and urbanization.

Security elements may also be behind an urbanization. For example, certain sensitive machines may regular change places so as to prevent having to deal with DDOS attacks. Similarly, a network may be cloned and, in the event of an attack, highly-authenticated clients are switched to the clone, whilst the original network is gradually deprived of its resources, to prevent it becoming a honey pot for the attacker.

Figure 2.16 shows a set of software networks, supplemented by diverse and varied virtual machines (servers, IP-PBX, Wi-Fi access points, etc.), which must obey an urbanization algorithm to optimize a set of criteria. Later, we shall revisit the question of the intelligence to be introduced into networks to optimize the processes of urbanization.

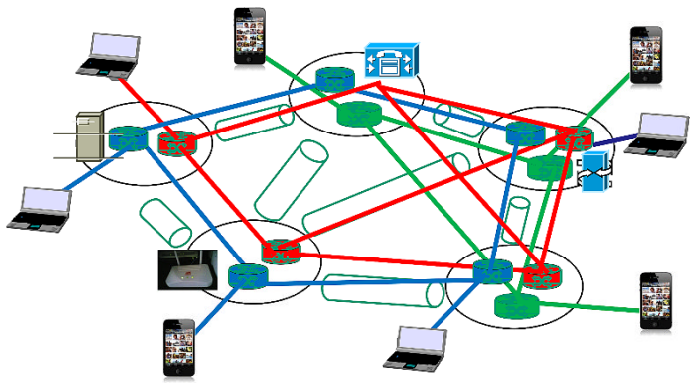


**Figure 2.16.** *The urbanization of a network environment. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

## 2.10. The NSX architecture

VMware is the first major company to have turned toward virtualization. It is a software company, not a hardware company, which explains why its NSX architecture is entirely software-based, and relies on the hardware of the network equipment manufacturers and on Cloud providers. The NSX architecture is represented in Figure 2.17. This architecture, as we have just said, is completely software-based and open, making use of Open Source software machines. The VMware platform is essentially based on the virtualization of the NSX architecture. This platform contains the main networking- and control elements. On the networking side, the machines may be level 2 or 3 routers or switches, depending on the user requirements. On the control side, the platform contains the basic elements with the possibility of putting virtual firewalls in place, managing the streams to channel them along paths determined by the load-distribution machine and by the opening of virtualized VLANs. (Note that here, the word "virtual" appears twice in "virtualized VLAN": indeed, a VLAN – Virtual Local Area Network – is a local software network that uses physical machines which may not necessarily be located in the same place, and that local network is, itself, situated in a virtualized universe).

The uppermost layer is formed of Cloud management systems, which may be either VMware or Open Source software such as Open Stack. The northbound interface is a REST-type API. The software network part is based on datacenters, with internal virtual devices or on physical machines from different manufacturers with which VMware has developed partnerships. The protocol of the southbound interface is OpenFlow, produced by Nicira, which was bought by VMware for over a billion dollars.

The hypervisors to support the virtual machines may be of any type, as may the physical hardware. The basic virtual device is Open vSwitch, which was also originally developed by Nicira. It is one of the very first virtual switches to have been available on the open source market. Today, it is an extremely powerful switch with numerous functions, which we shall now go on to discuss.

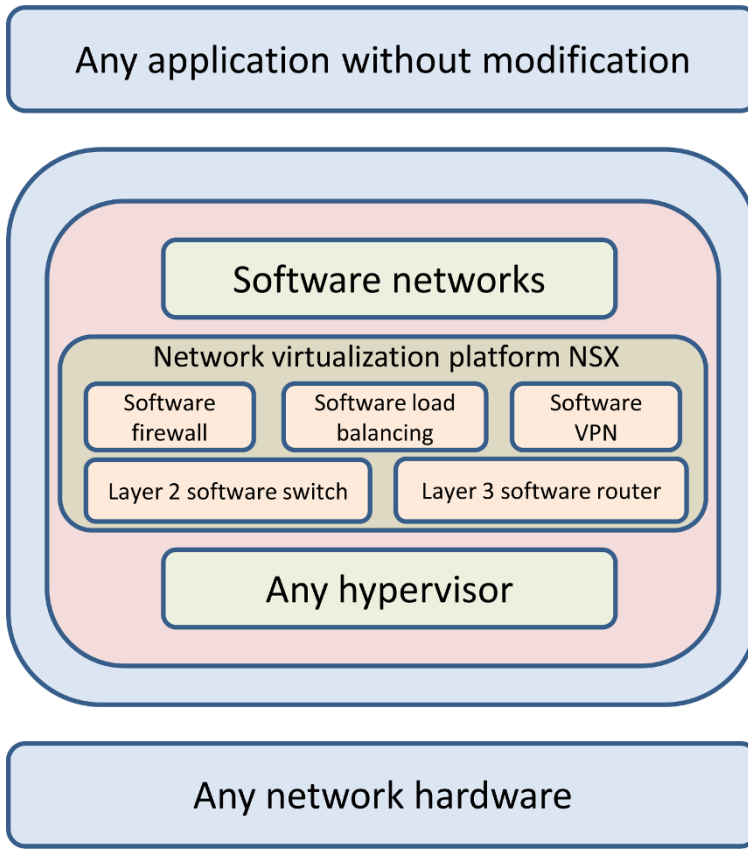



**Figure 2.17.** *The NSX architecture*

Figure 2.18 gives a somewhat more detail view of the NSX architecture with its various components.

Let us briefly come back to Open vSwitch, which represents one of the great *de facto* standards of virtual network machines. This open source switch is illustrated in Figure 2.19, with the different modules associated with its management. It enables us to group together virtual machines that manage security, quality of service, monitoring and automated control.

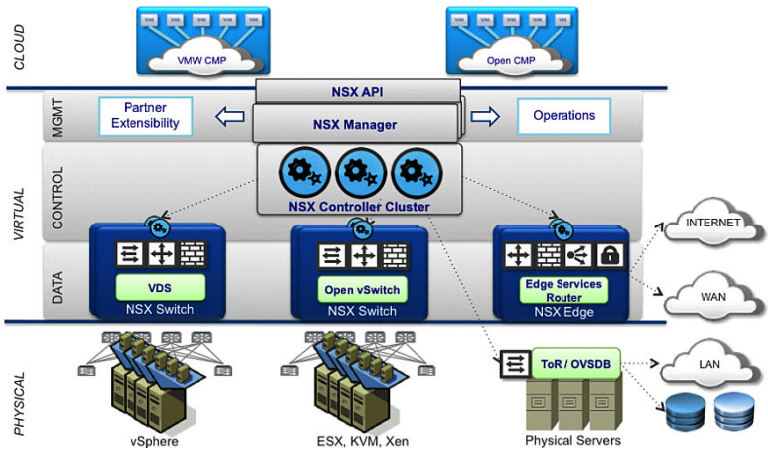



**Figure 2.18.** *Detailed NSX architecture*

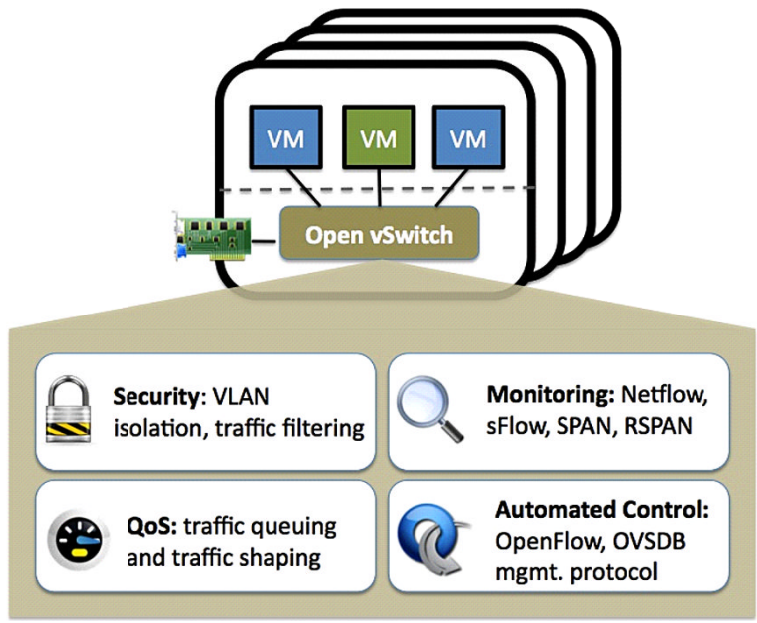



**Figure 2.19.** *The characteristics of Open vSwitch*

From the standpoint of security, the switch has VLAN isolation and traffic filtering. Monitoring takes place by way of various protocols such as Net flow or specific versions provided by manufacturers linked to VMware. Quality of service is handled by traffic-shaping and traffic-queuing techniques. The southbound interface with the controller uses several protocols – including, of course, OpenFlow, which is privileged, but also OvSDB (Open vSwitch Data Base) and various network machine configuration protocols.

The key functions of NSX VMware can be summarized as follows:

– logical switching: NSX performs complete level-2 or 3 switching in a virtual environment, dissociated from the underlying hardware;

– NSX gateway: NSX offers level-2 gateways for a transparent connection to the physical loads and the legacy VLANs;

– logical firewall: NSX opted for a distributed, virtualization-oriented firewall with identification and activity monitoring;

– logical load distributor: NSX provides a complete load distributor with SSL termination;

– logical VPN:NSX facilitates the opening of site-to-site or remote-access VPNs in software form;

– NSX API:NSX provides a RESTful API for integration with any Cloud management platform.

The mode of operation means that we can have complete dissociation of the physical infrastructure from the software networks formed of virtual machines. Virtualization of the networks enables us to superpose them on any physical hardware whose input/output ports are suited to the transmissions. In addition, NSX works with any type of hypervisor, which makes it easy to virtualize the servers. The NSX system facilitates the reproduction of the physical network model in software form, whether of level 2, 3, 4 or 7. NSX also includes automation by way of a RESTful (REpresentational State Transfer) API, which enables the Cloud management platforms to automate the supply of network services. Finally, NSX offers a platform which is capable of adding in services from other suppliers. These software and

hardware products, stemming from the integration of physical machines from partners of VMware, include network gateway, application provision and security safety.

## 2.11. CISCO ACI (Application Centric Infrastructure)

The company CISCO took rather a long time to come out with its own SDN architecture, which it did in 2014. This architecture has the application at the center. The objective is to automatically construct the networks needed for the applications in an environment which essentially comprises CISCO's own physical machines. This architecture is represented in Figure 2.20. As is indicated by the name, it is "application-centric", because the networks depend precisely on the characteristics of the service to be performed. However, it is also a hardware-based architecture, because CISCO hardware is needed in order to make it work, although physical machines from other manufacturers could be used to supplement the hardware part.

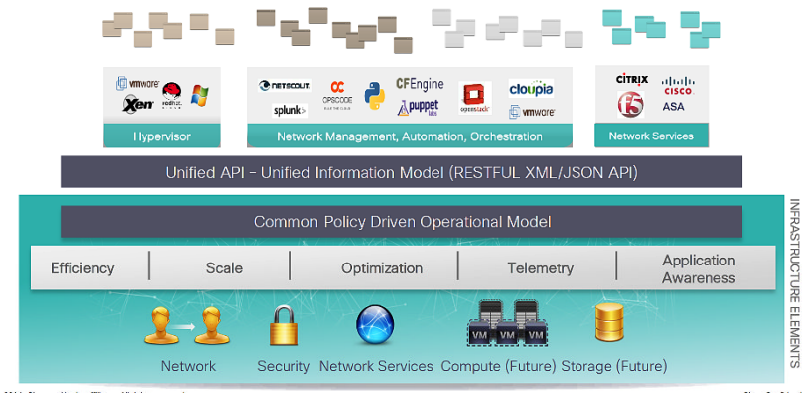



**Figure 2.20.** *The ACI architecture*

In this figure, we can see the heart of the architecture, which goes from the applications to a controller with numerous functions to create customized software networks. The northbound interface is particularly developed at CISCO. It uses a REST representation, like most SDN products. The Cloud supports the applications and – by

way of an orchestrator, which might be found, for example, in Open Stack – communicates with the controller using a Cisco API. The controller includes numerous intelligent modules to interpret the demands from the northbound interface and transform them into software networks associated with each application or set of applications with the same characteristics.

This intelligent module, or APIC (Application Policy Infrastructure Controller), is at the heart of CISCO's architecture. It uses artificial intelligence techniques to determine the relations between the needs of the applications and the networks which need to handle them. The complete ACI architecture is represented in Figure 2.21. We can see that it is based on a CISCO physical infrastructure in the form of the Nexus 9000 machine. The protocols for the southbound interface are of varying types, with no particular preference given to OpenFlow. The protocols between the servers and, more specifically, between virtual machines, are also of varying types, but they include TRILL and LISP, which we shall detail in the next chapter.

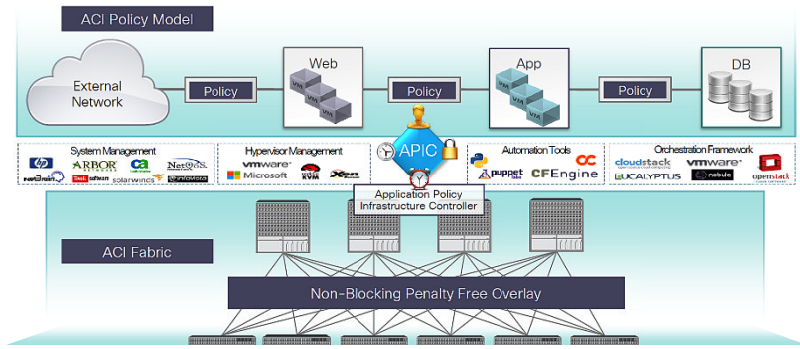



**Figure 2.21.** *Detailed ACI architecture*

Thus, the essential components of the ACI architecture are:

– the APIC (Application Policy Infrastructure Controller)made by CISCO, which is the main component of the architecture;

– the application sequencing profiles: the application sequencing profile corresponds to the logical representation of all the components

necessary to provide service and of their interdependence on the hardware;

– the physical devices necessary for CISCO's ACI architecture: they belong to the CISCO Nexus range. Cisco Nexus 9000 machines function either in Cisco NX-OS mode for purposes of compatibility and consistency with the older Cisco Nexus switches, or in Cisco ACI mode so as to take full advantage of the application services based on the policies and functions of infrastructure automation.

## 2.12. OpenContrail and Juniper

In 2014, Juniper also released its own SDN architecture, essentially based on open-source software, but with Juniper's own addition. The heart of this architecture is the OpenContrail controller.

OpenContrail from Juniper is an open, simple and agile SDN solution which is able to automate and orchestrate the creation of virtual networks on demand. Virtual networks are perfectly well suited for applications which require networks to be opened. In this architecture, we can emphasize its simplicity for the establishment of virtual networks which integrate smoothly with the existing physical networks, and are simple to manage and orchestrate. The environment is also open. There is no dependence on one particular provider. In addition, this open architecture eliminates the costs related to device manufacturers. The Juniper platform works with a broad range of open hypervisors, orchestrators and physical networks. Finally, an enormous advantage is the agility of the solution, which reduces the time to market of new services by automating the creation of software networks, facilitating the interconnection of private, public and hybrid clouds.

Service providers can use Contrail to offer a whole range of new and innovative services – particularly Cloud-based and virtualized services. The Juniper platform is shown in Figure 2.22. It should be noted that, as this figure shows, the platform tends to use Open Stack more than anything else, but this is not obligatory; other Cloud management programs are also acceptable.

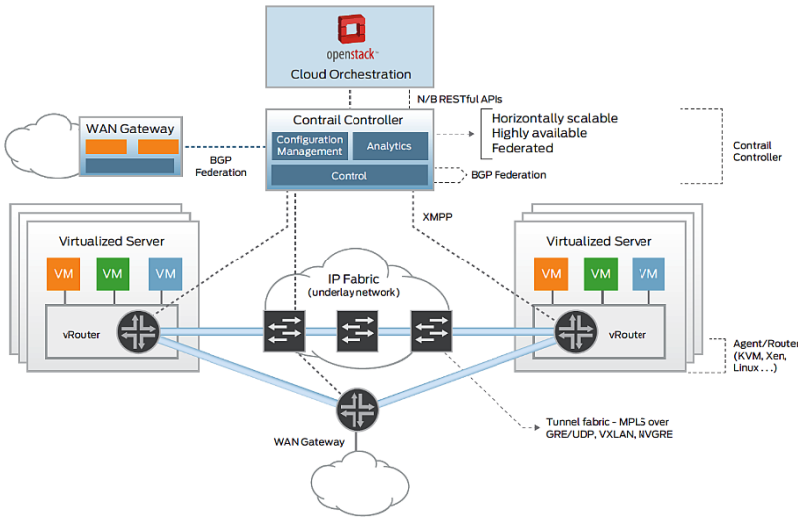



**Figure 2.22.** *The OpenContrail plat form, from Juniper*

## 2.13. Brocade

Brocade is another company which is investing heavily in SDN. They offer a platform which is more or less identical to that described above. It has a four-level architecture that is fairly typical in the world of SDN. The highest level pertains to the Cloud management system, which may be Open Stack, VMware Clouds or other solutions of the same type. This level contains the processes of orchestration and automation for requesting the opening of software networks.

The underlying control level contains the decision-making algorithms to determine which paths to open, and additional algorithms, linked to the costs, for example. In this category, we could point to cost-management algorithms which determine the most cost-effective placement of the paths, or those which will likely generate most revenue. The southbound interface can use different protocols, but OpenFlow and Open Script are favored for this relation between the controller and the software networks. The controller can be based

on Open Source technology such as OpenDaylight or come from other companies such as VMware.

The next layer in this architecture from Brocade is the virtualization layer, with a strong NFV component and the possibility of creating software networks on machines from Brocade or other manufacturers. This hardware part represents the physical layer. Various transport protocols are used in the physical network, including VxLAN, NVGRE, TRILL and NV03, which we shall examine in detail in the next chapter.

Brocade's solution offers flexibility and agility in the SDN architectures. It is illustrated in Figure 2.23.

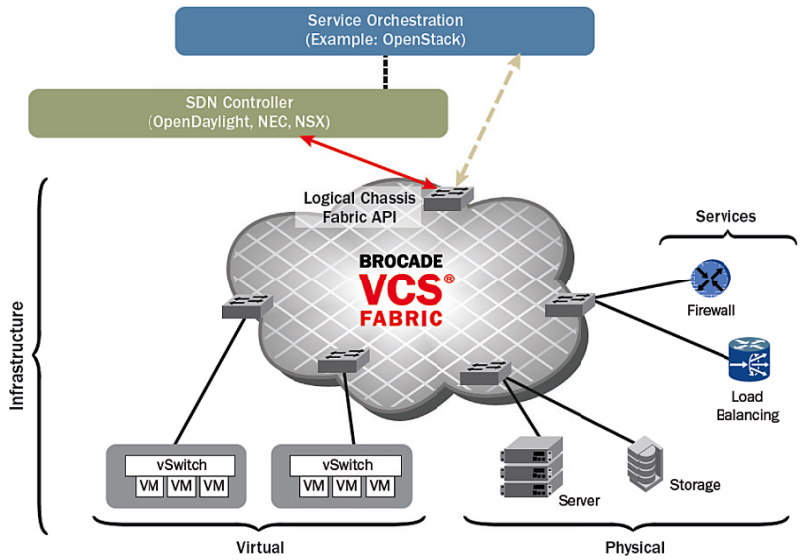



**Figure 2.23.** *Brocade's architecture*

## 2.14. Alcatel Lucent's SDN architecture

Alcatel-Lucent's solution is, to a large extent, attributable to its daughter company: Nuage Networks. The architecture is shown in Figure 2.24. The basic outlines are the same as those described above in reference to other manufacturers. The part where we find something

highly specific to Alcatel-Lucent pertains to optical networks and mobile networks. In particular, Alcatel-Lucent offers a vEPC (virtual Enhanced Packet Core) – i.e. a core network for a latest-generation mobile network, which can be seen as the first step toward a core network for 5G. With its products, Alcatel-Lucent wishes to encourage network functions virtualization, and to have a significant impact on the standardization of NFV.

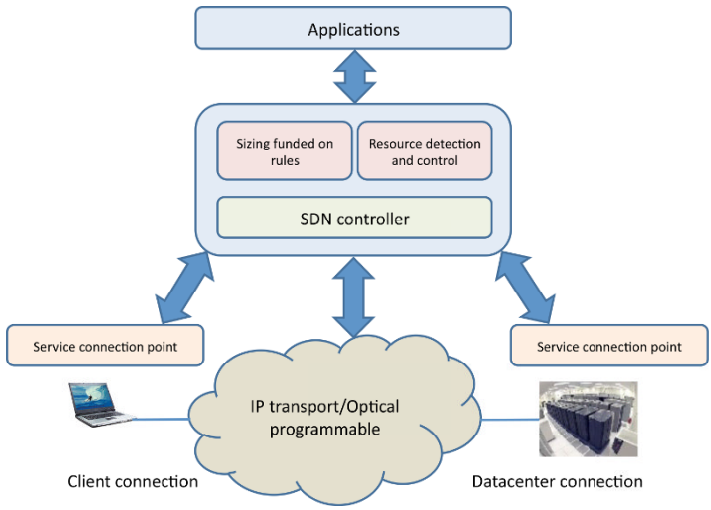



**Figure 2.24.** *The SDN platform from Alcatel-Lucent*

## 2.15. Conclusion

The new generation of networks is represented by Figure 2.25. It is presented in the form of datacenters of varying size, from gargantuan datacenters to femto-datacenters located very near to the user. These datacenters contain virtualized networking devices or networking functions that are decoupled from the hardware machines. This ensemble is controlled by pilot, orchestrator or controller machines. Figure 2.25 only shows one controller, but in fact, given the centralized vision of control, the number of devices handled by a single controller can be no greater than a few hundred, or in some cases, a few thousand. Therefore, we need a great many different controllers, which will be more or less mutually compatible. The

eastbound and westbound interfaces are likely to play an increasingly important role to facilitate the connection of the different sub-networks. It is clear that a very great many new propositions are likely to emerge in this direction, with extensions of protocols already in use, such as BGP.

**Figure 2.25.** *A view of SDN networks of tomorrow. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

One difficult problem which is beginning to be considered is that the urbanization of virtual machines which are in the datacenters. Depending on the optimization criterion, the time of day or the performance requirements, the virtual machines migrate so that they are located in the best possible place. Most typically, urbanizations are done for the purpose of saving energy, which tends to consist of grouping together the virtual machines on common servers, or for the purpose of optimizing performance, which involves the opposite operation – distributing the resources to the servers as much as possible.

It should be noted that in terms of networks, the optimum placement of the virtual machines is no longer an equal distribution between all the physical machines, but on the contrary, the filling of those physical machines with the maximum possible number of virtual

machines, obviously without causing a degradation in performance. The other physical machines are placed on standby to save energy. In the networks, this involves sending the streams along common paths, rather than dispersing them in the infrastructure network.

The SDN market is fairly difficult to predict, because whilst demand is high, so too are the costs involved in making the move to this new architecture. These high costs are due to novelty and to the difficulty in integrating SDN into the existing networks. In particular, operators who possess a significant physical infrastructure are wondering about the integration of SDN: how can it be introduced without an additional costly structure? Indeed, it is not an option to demolish everything and rebuild from scratch in this new context. Therefore, we need to introduce SDN in specific environments such as the renovation of a part of the network, or the construction of a new localized infrastructure.

Figure 2.26 gives an indication of what the SDN market could be in coming years. In this figure, we can see that there is a good distribution between service providers, the companies offering datacenter services, Cloud providers and, to a lesser extent, the companies' access networks.

Finally, if we look at the manufacturers' vision, we see two avenues, that are represented by VMware and CISCO: software-orientation and hardware-orientation. The software-oriented vision is installed on the physical machines of practically all manufacturers, whilst hardware-orientation is greatly favorable for manufacturers – especially well-established ones.

Although SDN is essentially intended for core networks today, there is a strong chance that it will become widely used on the periphery and that SDN will prove to be the right solution for controlling all peripheral devices. In particular, OpenFlow Wi-Fi access points are coming onto the market, and are likely to make their presence felt fairly quickly, with SDN following on in their wake. This is what we shall look at in Chapter 3: smart edges.

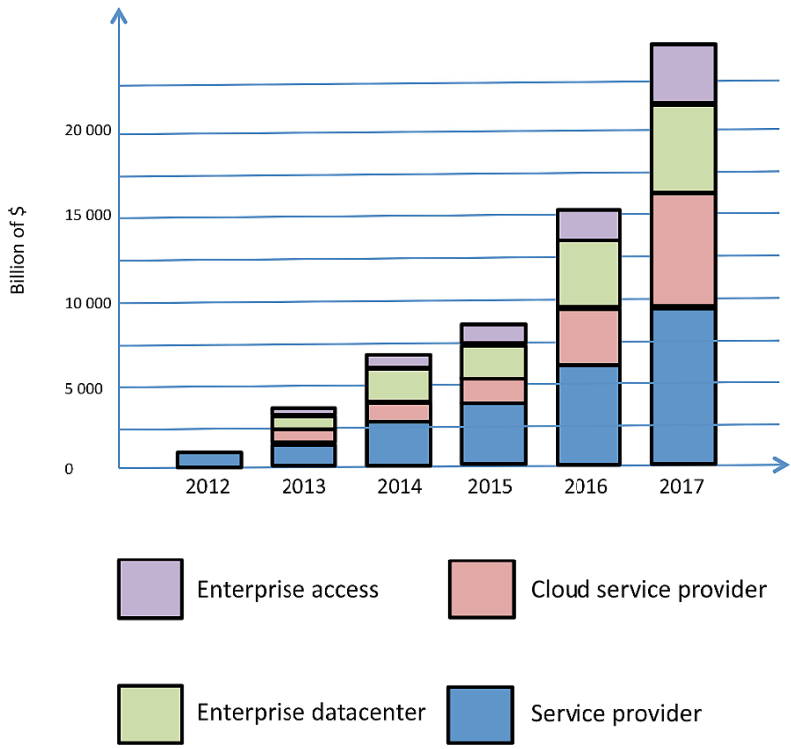



**Figure 2.26.** *The OpenFlow market, and more generally the SDN market. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

# 3

# Smart Edges

The first effect of the Cloud is to centralize the control and the orchestration system for the new generation of networks. Extreme solutions where there is only one center surrounded by radiating branches have been put forward as the solution of the future. For instance, we can cite the proposal of C-RAN (Cloud–Radio Access Network). However, many believe that centralizing everything is not the right solution, if only because of the problems of reliability and latency for real-time controls. The solution which actually comes into effect will undoubtedly need to be a compromise, with virtual machines situated at different points in the architecture. After explaining the different proposed architectural designs, we shall focus on intelligence in the extremities of the network, and thus what we call "smart edges".

## 3.1. Placement of the controller

Figure 3.1 illustrates the different scenarios for the placement of the controller in a software network environment.

In Figure 3.1, we see three main scenarios. The first is highly centralized around an SDN controller. The control data are fed back automatically to that center. The second solution is to use intermediary machines at local or regional levels. These machines could be SBCs (Session Border Controllers) – a term which tends to be used in operator networks – or NACs (Network Access Controllers), corresponding to the solution favored by companies. In

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

this scenario, the solution sends the control data to local or regional datacenters. Finally, the last example of architecture consists of positioning the control as close as possible to the user in femto-Clouds. The advantages are clearly the very short response time, the agility of the solution because of the possibility of changing the virtual machines used in the femto-datacenters and the addition of a form of network intelligence very close to the user. The disadvantage here is that the solution involves widespread distribution of the information, with all the complications that this can entail to perform controls, which, obviously, will not take place locally.

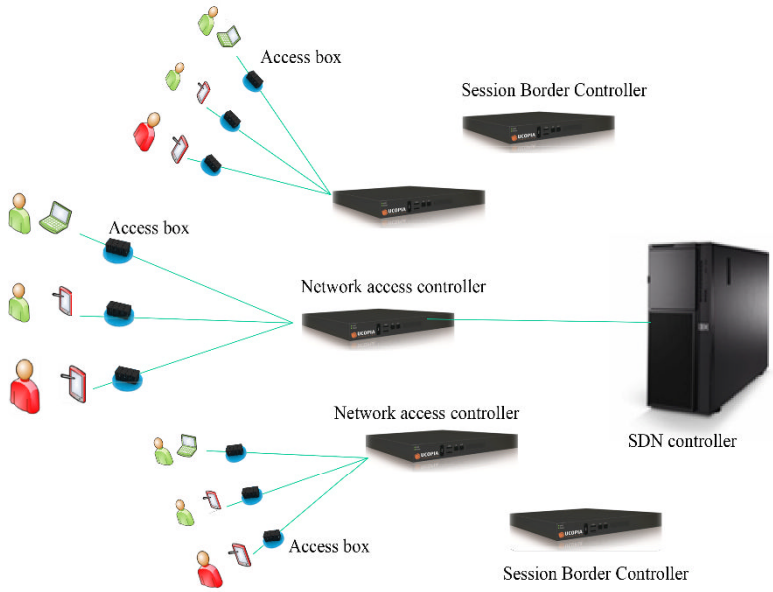



**Figure 3.1.** *Scenarios for the placement of a controller*

The three main solutions described above can be combined, with two levels or even all three levels, acting simultaneously, certain virtual machines may be migrated to the femto-datacenter, whilst others remain in a central or regional datacenter.

If we examine the first scenario, it represents a highly centralized system on a large Cloud with several high-capacity datacenters. These

datacenters are usually situated a long way from the periphery. This solution, which has been put forward, amongst others, in the context of C-RAN, is a revolution in the world of networks. Indeed, all the computations are performed in a powerful Cloud, with the decisions being centralized. This solution requires enormous datacenters, usually situated far from the periphery. This case is illustrated in Figure 3.2, in the form of a centralized C-RAN. We shall look at this proposed solution in greater detail in Chapter 4, but right now we are going to examine a few of the fundamental issues.

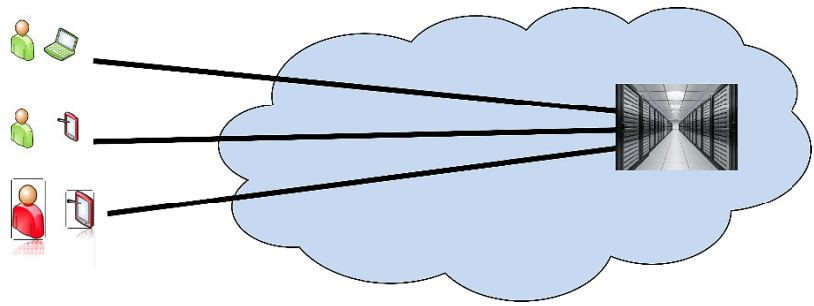


**Figure 3.2.** *Centralized C-RAN*

The revolutionary aspect is Cloud access network, which is concentrated in an optical network known as RoF (Radio over Fiber). The local loop, in this case, is formed of greatly-simplified antennas which do not process the signal, but merely forward electromagnetic signals to the center over an optical fiber. The signals reach a datacenter with virtual machines to process the radio signals. It should be noted that this solution enables us, with a single antenna, to process almost any type of signals received on that unique antenna: Wi-Fi, 3G/4G/5G and Bluetooth, which is a significant innovation for the future.

The datacenter, be it large or small, can then receive all the virtual machines necessary for the computing of the applications. This solution requires a physical star-shaped network using optical fiber technology, where there are no longer any concentration- or control

devices. There are no intermediary boxes or computing machines. The signal is sent to the central datacenter exactly as it is. In other words, there is no longer a network, i.e. no messages, no packets, no frames – just a succession of radio signals.

In terms of cost, the centralized C-RAN system is certainly one of the most affordable solutions on the market, because all the intermediary machines are eliminated. However, this solution exhibits the disadvantage of being centralized, and therefore requires all the usual precautions for centralized environments to be exercised. For example, the amount of resources in the different datacenters needs to be doubled or even tripled. Another problem is the response time for simple or local actions, such as inter-cell handover. If we assume there is a distance of 300 km between the periphery and the center, the propagation time there and back represents around 2 ms. In addition to this propagation time, we also need to allow for the time taken to run the control algorithms, and therefore reaction times of around 5 ms. Most control applications can cope with values like these. However, they are too long for simple actions and, above all, they are too costly in terms of overall data rate.

The solution represented by the centralized C-RAN architecture is certainly interesting, but there is a danger that it will be rejected in developed countries whose infrastructure is ripe for a decentralization of the datacenter. On the other hand, for developing countries where there is very little infrastructure, the C-RAN represents a good opportunity to gain direct access to 5G environments at a low cost.

This centralization may be compensated by a slight distribution on the periphery, in which case we see the emergence of small Clouds (known as *Cloudlets*) situated near to the relay antennas. This solution is shown in Figure 3.3.

The second solution for the placement of the controllers is to build them into regional Clouds. In this case, the controllers are situated at the level of the company, or the DSLAM (Data Subscriber Line Access Module), in a telecommunication network. Here, the controllers only manage networks of limited size. This solution

requires interfaces between controllers (westbound interface for heterogeneous controllers and eastbound interface for identical controllers). In addition, this solution does not eliminate the local loop, which therefore resumes its original duties. It merely handles the connection, which is done in signal mode, with framing and packeting being done in the Cloudlet. We shall revisit this solution, which is popular in developed regions, later on.

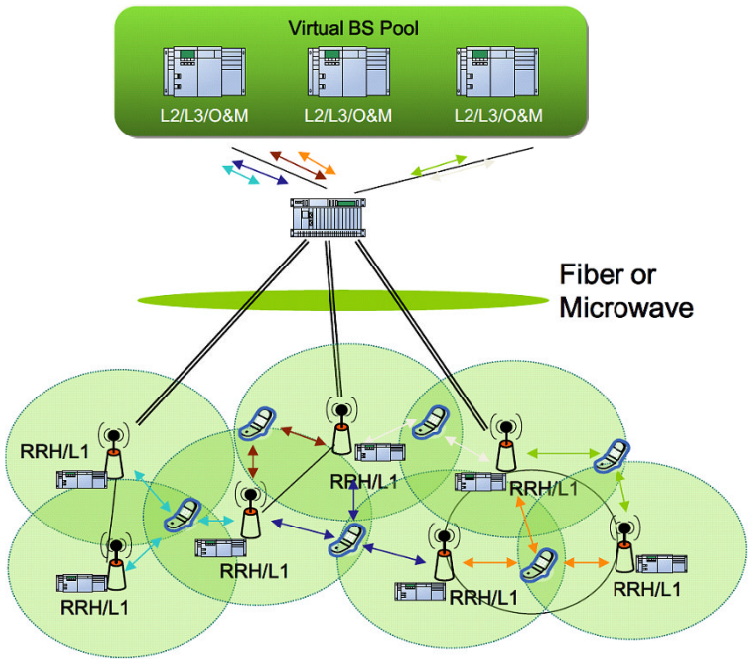



**Figure 3.3.** *C-RAN with a distribution element*

The control system is partially distributed, and coordination is needed between the different controllers for all communications between two different sub-networks. The eastbound and westbound interfaces become particularly important in this context. Figure 3.4 presents this architecture, wherein the controllers are encapsulated in regional datacenters.

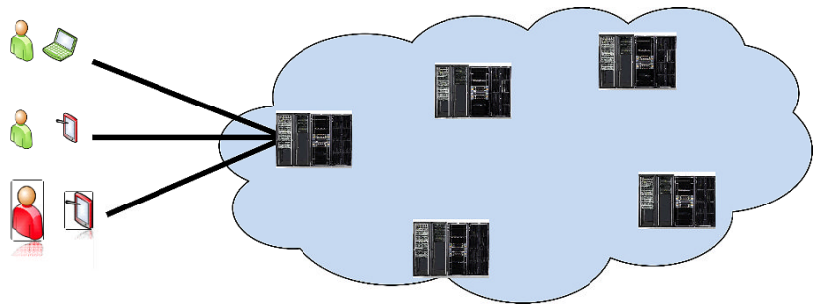


**Figure 3.4.** *Cloudlet solution*

Looking again at Figure 3.3, it seems to suggest that Cloudlets are available even nearer to the user than regional or local centers. Cloudlets are situated in the access antennas. This represents another solution which we shall examine in detail in this chapter: that of femto-Clouds.

The third scenario, which best represents a smart edge, involves total decentralization to femto-Clouds. Those femto-Clouds are associated with femto-datacenters located very close to the clients, i.e. in the access points or in boxes situated in the user's area. The immediate advantage of femto-Clouds is the extremely short response time to run a control algorithm, and therefore the ability to carry out operations in real time. However, this requires devices with significant amounts of resources – and many of them. Given the reduction in the price of hardware, this solution is increasingly gaining favor in many services offered by manufacturers. In addition, there is a need to install boxes to accommodate the femtocells, and it is possible to take advantage of this to enlist a little local power, running virtual machines to establish connections and perform local computation in those boxes. In this case, we speak of Metamorphic Networks, or MNets: the network metamorphosizes depending on the client who is connecting.

In Figure 3.5, we have illustrated an MNet configuration with an MNetBox (Metamorphic Network Box) from VirtuOR

(a start-up of University Pierre et Marie Curie). The developments undertaken by VirtuOR, amongst others, pertain to this new generation of metamorphic networks, where each client is allocated his/her own network. The network appears differently for each client, because it metamorphosizes depending on the client connected. The MNetBox is, in fact, a femto-datacenter in which the clients' virtual machines are located, to handle the networking applications (Networking Apps). These networking Apps include all network-level applications that may be found in the literature and in Clouds, such as IP-PBX, routers, switches, SIP servers, firewalls, etc., in particular the virtual machines created by NFV (Network Function Virtualization) standardization, such as DPI (Deep Packet Inspection), computing of NAT, BRAS, etc.


**Figure 3.5.** *A network with a femto-datacenter*

## 3.2. Virtual access points

A femto-Cloud may support diverse virtual machines – particularly virtual Wi-Fi access points. The client has a choice as to which access point he/she will use: it may be their office access point, their operator's access point, their home access point or indeed any other access point with its own individual characteristics. The box may contain a large number of virtual access points – all different from one another, and of course, isolated. The exact number depends on the power of the femto-datacenter and on the number of potential users. The data rate received by each user is dependent upon the number of virtual access points, the number of users on each virtual Wi-Fi access point and the Wi-Fi technology used. The virtual access points share the physical antenna, using a token-based technique, for example. Obviously, the capacity of each virtual access point is contingent upon

the number of tokens, and is therefore not as powerful as a non-shared access point. However, in light of the rise in Wi-Fi power, as we shall see in Chapter 6 on 5G, each virtual access point can have sufficient capacity. In particular, the use of the IEEE 802.11ac standard means that each virtual access point is able to achieve sufficient data rates for all conventional applications, and even some less-than-conventional ones.

Undeniably, there are some flaws with virtual Wi-Fi access points! Each virtual Wi-Fi access point has its own signaling frames. If the number of virtual Wi-Fi access points is high, then the overhead expenditure is great.

The access point described here is a virtual Wi-Fi access point, but it can perfectly well be a virtual Node-B, i.e. the device which manages the physical antenna for a mobile network. In this case, the box may be considered as an HNB (Home Node-B) or an MNB (Metro Node-B), extended with significant capacities to receive virtual computation machines.

More generally, all wireless access techniques can be virtualized, and thus facilitate personal connections. Depending on the hardware that can be replaced by software, major advances can be implemented to work toward multi-technology access solutions.

In Figure 3.6, we show the context of virtual access points, which is one of the fundaments of the “smart edge” environment. Indeed, for users to connect, we need to put antennas in place, and we can exploit those antennas to do all the local computation needed for access control. In this context, a large portion of the network’s intelligence is on the periphery. The difficulty is in handling communications between users situated in different sub-networks, because we need to coordinate the networking elements which are used for the end-to-end communications. Communications through the eastbound and westbound interfaces are crucial for SDN solutions, to interconnect the femto-datacenters.

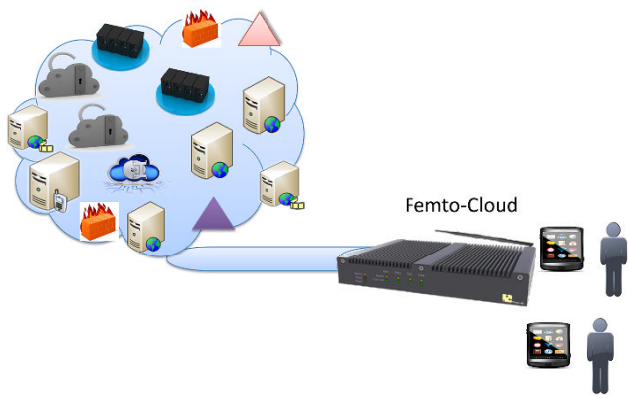



**Figure 3.6.** *Context of a femto-Cloud network for a "smart edge"*

Figure 3.6 shows an example of a femto-Cloud, serving two virtual access points, capable of connecting two clients using different network operators. Each client has, in the box, his/her own virtual Wi-Fi access point or dedicated HNB to connect to the network. The two clients do not know one another and they each have their own personalized access point. Plainly, other virtual machines can be placed in the box to manage a particular interface or perform a specific signal computation operation.

More generally, operators are beginning to look at the possibility of rolling out their own virtual access point so as to take position in places where their clients may be. For example, a lecture hall which has a femto-datacenter could connect all the clients of the operator X after migration of a virtual access point X into the femto-datacenter. The virtual access points belong to the network X and are managed directly by the operator X. If the physical access point breaks down, the responsibility for its repair lies with the infrastructure operator, e.g. the owner of the lecture hall. There are various business plans to provision a femto-datacenter, ranging from renting resources to install a virtual machine to charging by the amount of time and capacity used.

## 3.3. Software LANs

Looking again at the examples from the previous section, it is possible to describe a software LAN. This is a local-area network formed of different femto-datacenters in which the virtual machines can be stored in memory to create these software LANs. Numerous virtual machines such as routers, switches, firewalls or application machines can be transported to the femto-datacenters from a central Cloud or a local server. The decision to leave a machine in the central or regional Cloud or to transport it to one of the femto-datacenters depends on the characteristics of the application and on the evaluation of the traffic that it could generate.

Figure 3.7 illustrates a scenario for a company that has several boxes to create software LANs. In this case, to begin with, the company's Wi-Fi access points need to be transported to all of the boxes, or else we need to select a number of virtual access points corresponding to logical networks which, themselves, will be constructed to serve specific applications, such as VoIP, IPTV, bank access, messaging, file transfer, professional application, etc. The software networks are established by the virtual machines that run in the boxes. It is also possible to add a completely isolated network for visitors to the company. This allows guests and visitors to connect to the Internet, using the company's network, with no risk of an attack.

The software networks can use all different protocols from one another, including IPv4, IPv6, MPLS, Ethernet, SDN, etc. It is necessary to choose the protocol which best suits the application for which the data are being sent over the network.

The connections between the boxes may be hardwired or wireless, using protocols corresponding to the type of network chosen by the user. This may be OLSR, AODV, TRILL, or indeed any other protocol developed in this field.

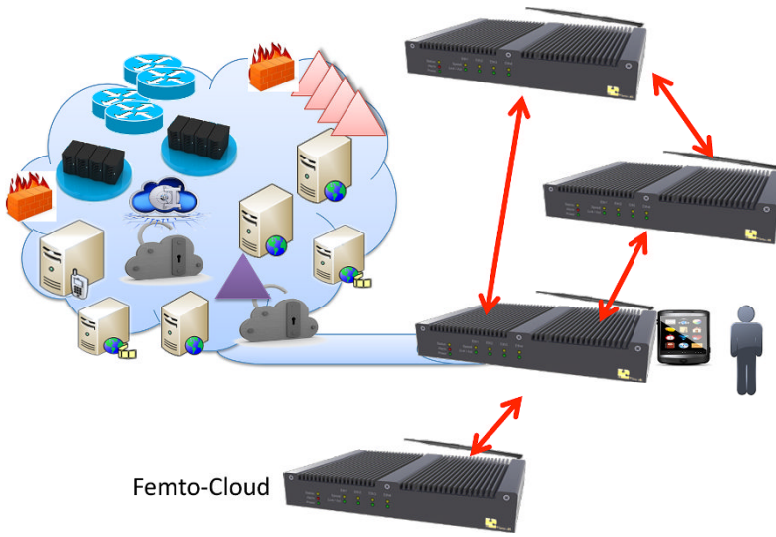



**Figure 3.7.** *A femto-datacenter environment to create virtual LANs*

Software LANs open up new possibilities, such as the installation of SDN Wi-Fi access points. Indeed, by combining a virtual Wi-Fi access point and an SDN switch such as Open vSwitch, it is possible to send Open Flow commands from a controller to the Wi-Fi access point. It is also possible, of course, to install an SDN controller such as Open Daylight, Open Contrail or Flood Light in one of the boxes to control the other boxes making up the software network. In this scenario, as in the previous ones, certain virtual machines may remain in the central or regional datacenters, depending on their size and their rate of use.

Although it is predicted that SDN using central networks will become dominant, an ever-growing number of networking individuals are calling for the implementation of SDN in the periphery, because this way the control mechanism is much simpler, the cost is much lower than WAN (Wide Area Network) solutions offered by device manufacturers and the system is just as agile.

Figure 3.8 shows that the control virtual machines may be located in datacenters situated at different levels. The controllers are

virtualized in datacenters present at the central or regional level, but can also be found in the femto-datacenters situated on the periphery. Indeed, the Wi-Fi access points (Nodes-B or e-Nodes-B) require hardware, and must be close to the user. These boxes also facilitate the easier introduction of 5G, which we shall look at in more detail in Chapter 6. Indeed, the Internet of Things requires new interfaces that are well suited to the "things" in question. These interfaces can be handled by specially-designed virtual machines.

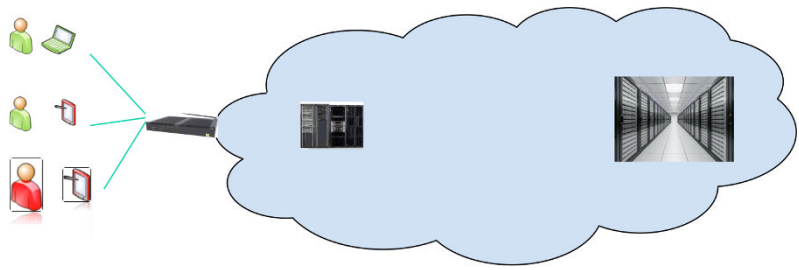


**Figure 3.8.** *Hierarchy of controls and datacenters*

## 3.4. Automation of the implementation of software networks

An important point in the advent of the new generation of software networks, whether in terms of the core network or the LAN, relates to automation for the establishment, management and control of personalized networks designed in the context of software networks, which may be SDNs or use legacy architectures, or indeed a mix of the two, given that each software network is entirely independent of the others. With this in mind, as indicated by Figure 3.9, we need to add an auto-piloting system to the general architecture. The basic domains to be piloted are the servers, storage facilities and networks, but we also need to add security and management elements, which have become absolutely indispensable for the proper operation of a complete system.

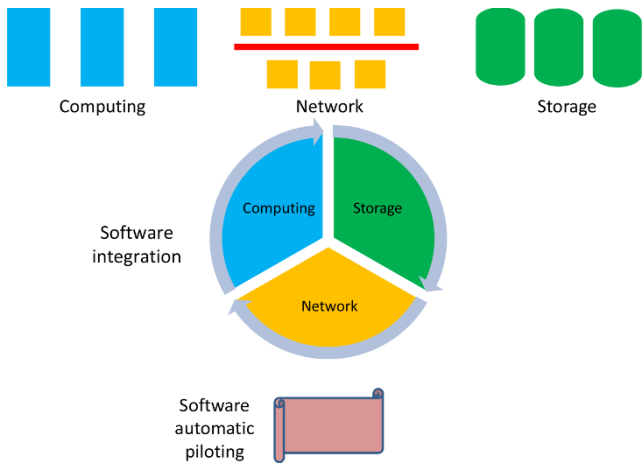



**Figure 3.9.** *Self-piloting system. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

The piloting system may be referred to by a number of different names, such as "orchestrator", "governor" or "high-level pilot", to indicate that this element directs the controllers which are situated in the control layer. This piloting system feeds the underlying layer with the best possible information, and the control layer makes the decisions on the basis of the information received from the pilot.

The piloting system may be centralized, which is generally the case with centralized Cloud architectures, or totally distributed to the controllers, or, as is most often the case in a conventional situation, a mixture between centralized and distributed organization. At present, numerous developments are being made in this field, with virtual machines that act as intelligent agents.

## 3.5. Intelligence in networks

Intelligence is a much-used term in computer science, which simply means the ability to communicate, to reason and to make decisions. Up until the start of the 2000s, intelligence in networks was very low. The concepts of intelligent networks, dating from the start of the 1990s, introduce a primary intelligence, whose role is to

automatically adapt the network components to the users' demands, but without reasoning, merely based and predefined rules. The next section is devoted to this view of intelligence in networks. It also introduces autonomic networks, which replaced programmable networks or active networks. An autonomic network is one which is capable of configuring itself, and wherein the nodes may become autonomous in case of failure or cutoff in communications.

Since the beginning of the 2000s, true intelligence – i.e. reasoning-based intelligence – has been offered by certain network components, which are able to make decisions pertaining to control or management. The devices which make those decisions stem from the domain of artificial intelligence and smart objects. In particular, multi-agent systems have been around for a long time, able to handle security and failures.

Intelligent agents offer the first category of tools whose introduction on a large scale could modify management and control environments, making them more autonomous and more reactive.

We shall now examine the reasons for this power, and then the way in which we construct multi-agent systems.

## 3.6. Management of a complex environment

As networks have become increasingly complex, the management and control of these environments has become necessary for a variety of reasons, which we shall now go on to examine. Network environments are becoming increasingly dynamic, as we have seen. Numerous and varied applications are interwoven, making it difficult to control the resources. The statistical gain – i.e. what we gain by treating the data packets statistically – in packet-transfer networks is undeniable, but, if the data flows too far exceed the network's capacities, a meltdown of performance is inevitable.

Network environments are, by nature, distributed, although the trend is toward centralization, which makes it complicated to control and manage them. In addition, enormous scale necessitates even

closer control. Sizing is a complex problem to grasp, and we can state that there are no truly effective tools available in this domain, in that the parameters that need to be taken into account in a complex network environment are difficult to fully appreciate. We have a choice between data rate, response time, degree of usage of the line couplers and of the central units, bit error rate, packet error rate, repeat rate and failure rate. In addition, the values of the mean, the variance and sometimes the higher-order moments need to be taken into consideration in order to gain a real idea of the performances.

The engineering of network design involves two main aspects: the qualitative and the quantitative. The qualitative aspect often corresponds to operational security, in the sense that it is possible to prove that the system is stable or that there is no state in which the network ceases to work. The quantitative aspect refers to the values of the parameters listed in the previous paragraph, with the aim of quantitative analysis being to show that these values are reasonable for normal operation of the network.

Security is an important function, to which intelligence may contribute. Today, a certain degree of standardization enables us to have a clearer view of the problems, and a few major classes of security have been defined, corresponding to needs clearly expressed by the users. We can easily imagine the contribution made by intelligent tools in the world of security, to discover anomalies, analyze them, give a diagnosis, propose a solution and resolve the problem.

Management is also a domain where intelligent agents could play a leading role. When a network is running, it needs administration – i.e. we need to be able to control all the operations that take place in the network, from breakdowns and accounting to security, performance management and username management.

Various specific administrative domains already use intelligent components – in particular, the following:

– configuration management;

– security management;

– fault management;

– performance management;

– accounting management.

Intelligence of the agents may stem from different areas. The most usual model stems from the domain of distributed artificial intelligence, or DAI.

Artificial intelligence means that a device can take the place of a human being to perform a task. DAI is equivalent to a society of autonomous agents, working together to achieve an overall objective. There are numerous reasons to turn to DAI – notably the following:

– integration of different points of view. When the data become more specific, inconsistencies may occur in the rule base. The ways in which knowledge is expressed are different depending on whether we are addressing the user, the developer or the technician. Also, two experts who have the same expertise will not always come to the same conclusion. The different points of view are also often contradictory: one might attach a great deal of significance to the costs, and therefore favor a less expensive system, whilst another would choose to develop the publicity, and thus opt for a dearer system. The use of DAI helps to achieve a compromise between different options, by negotiation;

– representativeness of the real world. In general, it always tends to be a group of experts, with different qualifications and specialties, who work together to realize a set goal. In addition, whilst it seems easy to understand and therefore to model the behavior of individuals (all of their exchanges) thanks to the numerous sociological studies that are available, the way in which the human brain works, and the reasoning process, are less well understood.

For these reasons, the application of distributed artificial intelligence is gradually becoming a reality in the context of network management.

## 3.7. Multi-agent systems

An agent is an autonomous entity, capable of communicating with other agents, and of perceiving and representing its own environment. A collection of these agents, interacting with one another, forms a multi-agent system. We classify such systems according to numerous criteria, such as the size of the agents, the number of them interacting, the mechanisms and the types of communication, behavior, organization and control of each agent, the representation of the environment, etc.

Based on these criteria, we distinguish two main categories of multi-agent systems:

– systems of cognitive agents;

– systems of reactive agents.

Cognitive agents have an explicit representation of the environment and of the other agents. They are able to take account of their past, and operate with a social means of organization. Systems using this type of agent will have only a small number of agents. Several levels of complexity can be envisaged:

– processes in which the actors implement communication directives;

– communicative modules, which use specialized communication protocols (requests, commands);

– cooperative agents, which work with the concepts of skill, mutual representation and task allocation;

– intentional agents, which use notions of intention, commitment and partial plans;

– negotiating agents, which carry out conflict-resolution by negotiation;

– organized agents, which act in accordance with regulation and social laws.

The agents communicate with one another using a specific language. This is intentional communication, which essentially comprises two types: communication by information-sharing and communication by message-sending.

Communication between agents takes place by information-sharing when the solution to the problem is centralized in a global data structure, shared by all the agents. This structure initially contains the data of the problem, and is enriched over the course of its resolution until the solution is reached. It constitutes the only means of communication between the agents.

This type of communication is often spoken of as the blackboard model, discussed in detail in numerous publications. The agents deposit and read a piece of information in a shared data zone – the blackboard – as illustrated by Figure 3.10.

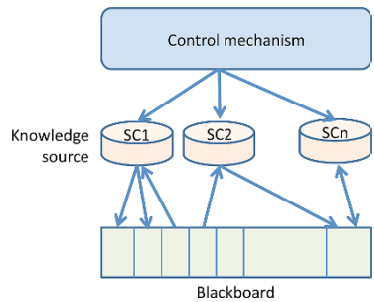



**Figure 3.10.** *Operation of the blackboard*

A blackboard system comprises the following three basic elements:

– the blackboard, in which all the elements involved during the resolution are stored. This data structure is shared by the agents and is organized hierarchically, which enables us to considerer the solution at various different levels of detail;

– the agents, which generate and store their data in the blackboard. They are independent modules, referred to as “knowledge sources”. Their role is to cooperate to solve a given problem. The knowledge sources are independent, as they are not aware of one

another's existence, and merely react to the events of the blackboard being changed;

– a control device, which ensures that the system operates in accordance with a certain strategy. Its role, amongst other things, is to resolve conflicts of access to the blackboard between the agents, which may intervene without having been triggered. Indeed, in the absence of centralized control, the knowledge sources react in an opportunistic manner – i.e. they react as best they can. This control device itself functions in accordance with the blackboard model.

Blackboards have the advantage of providing structure and an automatic method (divisions and hierarchy) in the way in which we approach a field of knowledge. They also exhibit the advantage of organizing sets of rules in systems with production rules. However, their lack of local memory means that they are not able to truly function as multi-agent systems. As a general rule, multi-agent systems use a blackboard for each agent.

Multi-agent systems based on message communication are characterized by total distribution of the knowledge, the partial results and the methods used to achieve a result (see Figure 3.11). Certain actor languages offer an accurate incarnation of this type of system. Communication can take place either on a point-to-point basis or by broadcast.

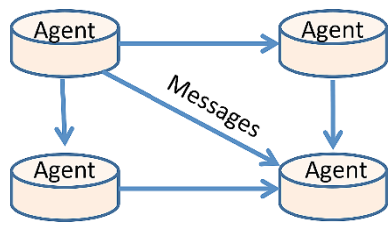



**Figure 3.11.** *Operation of a multi-agent system*

Such a system is built around two components:

– *local computation*: quite unlike blackboard-based systems, the knowledge is no longer concentrated in the same space, but instead is

compartmentalized and distributed between the different agents. An agent can only manipulate its own local knowledge base, send messages to other agents that it knows (its "acquaintances") and create new agents. At any given time, the agents do not have an overall vision of the system, and only have a local point of view regarding the elements;

– *message forwarding*: when an agent sends a message, it specifies which agent the response to the message should be sent. It may indeed be the agent which sent the original message, but it may just as well be another agent, specially created for the circumstance.

The agents have a more or less precise knowledge of the other agents in the system. They must be aware of and represent the abilities of those agents, and the tasks being performed at any given moment, the intentions and commitments of the agents. This aspect of matters raises the problem of the representation of this type of knowledge, and also of its updating.

The principle of task allocation constitutes one of the essential points relating to multi-cognitive-agent systems. A problem is composed of a certain number of tasks, performed by agents which bring together all of the partial solutions to obtain the overall solution (see Figure 3.12).

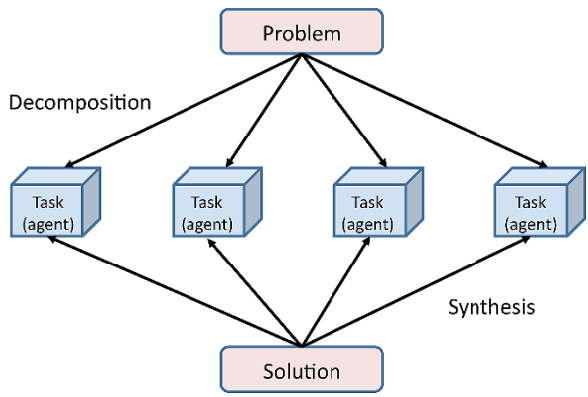



**Figure 3.12.** *Problem-solving*

To successfully allocate the tasks, we need to know the skills of each of the agents, decompose a problem into sub-problems, distribute the solving activities to the appropriate agents and, as required, redistribute those activities dynamically.

Task allocation models can be classified into two categories:

– *centralized allocation*: in this modular approach, an agent breaks down a problem into sub-problems and distributes those sub-problems to other agents, which are subordinate to the original agent. In this case, the actions are consistent, but there is a problem of reliability and extensibility. In addition, an agent is entirely dedicated to the distribution of tasks. Thus, the system is not used to the full extent of its capabilities;

– *decentralized, or distributed, allocation*: each agent is capable of breaking up its own problem into sub-problems and thus distributing the tasks associated therewith. All the agents carry the same weight in the decision-making process. This type of allocation is appropriate for applications which already have a distributed structure. The reliability and the possibility of extension are better than with the previous model, but it is more difficult to maintain consistency.

Three types of methods may be employed to decompose a problem:

– static: with each agent knowing the skills of the other agents, the sub-problems can be attributed to the best-qualified agents;

– dynamic: the agents work together to distribute the sub-problems in the most effective way possible;

– mixed: each agent is aware of the skills of the other agents, but this knowledge is periodically updated.

The autonomy of the agents is founded on the concept of intentionality. We can differentiate intention in an action from the intention to commit an action in the future. In the latter case, we have a standing goal. In order for an agent to have the intention to perform an action, that agent must believe that the action is possible, envisage committing to performing it, estimate that if certain conditions are

fulfilled then the action can successfully be carried out, and finally, not attempt to bring about all of the consequences. However, we may well wonder what happens when the action has been carried out by another agent, when an agent has two intentions in the conditions in which an agent can give up on its intention.

## 3.8. Reactive agent systems

A reactive agent does not have an explicit representation of its environment, and cannot take account of its past. The way in which it operates is simple, following a set of pre-programmed stimulus/response-type decisions. The system's organization is biological, and the number of agents present in such a system is very high. Communication is non-intentional. For example, the agents leave traces of their presence, or signals, which can be perceived by other agents. We then speak of environmental communication.

This type of agent results from the following postulate: the interaction of a large number of simple agents can emerge from complex organizations.

We can consider various levels of complexity for a reactive agent:

– stimulus/response: simple reactions to the environment;

– coordination of elementary actions: inhibition mechanisms, relations between elementary actions;

– reactive cooperation: mechanisms of recruitment between agents, aggregation of elementary agents;

– reproduction: mechanisms of reproduction of reactive agents;

– organization of reactive agents.

Eco-resolution is a problem-solving technique based on the use of reactive agents. Here, problem-solving is considered to be the result of a set of interactions. This view stands in opposition to those adopted in conventional approaches to problem-solving, such as space state exploration, which poses problems of combinatorial explosion.

The distributed problem-solving approach is based on a radically-different idea: that of the appearance of configurations as stable or steady states of a dynamic system, whose evolution is due to interactions stemming from the behaviors of small, fairly simple agents.

In conventional systems, all the data are contained in the statement, with the system bound to find how to pass from the initial configuration to the final state. On the contrary, with the phenomenon of eco-resolution, the determination is purely local. The agents are characterized by behaviors of satisfaction, attack and retreat. The problem itself is defined by a population of autonomous agents, all seeking satisfaction. The final result is the consequence of a non-deterministic interaction. The model defines the combination of behaviors.

In conclusion to this section, we can argue that the study of learning, real-time operations or distribution, corresponds to a need in the area of network management. However, it is the last point which seems to hold the greatest interest, in that multi-agent systems, in a manner of speaking, constitute a generalization of the techniques of expert systems. Thus, they provide added value to conventional AI systems by offering a new type of architecture, involving communication as well as internal reasoning. The aspects of opening and distribution make them interesting for a network management system.

As regards the real-time aspect, the approach adopted by AI seems less clear. Nevertheless, it is a crucial step in the design of a network administration system. Indeed, the response times (to a failure, for example) must be as short as possible, although this criterion is not as important here as it is in decision-support or command-support systems.

Learning remains the Achilles' heel of knowledge-based systems. As long as these systems are unable to improve, expand and refine their knowledge on the basis of their own experience, they will be dependent upon the goodwill and availability of experts and the quality of a manual update.

## 3.9. Active networks

Active networks are similar to other packet- and frame-transfer networks. The basic unit transferred in these networks is the packet. The role of the nodes is to examine the different fields in the packet, which are placed in accordance with a very rigid predefined system. In particular, the address field determines the output port. The different fields in the packet are interpreted by a virtual machine. We can consider that this virtual machine is a packet interface, which is typically called a "network API", or "NAPI" (Network Application Programming Interface). For an IP network, the IP API is the language defined by the syntax and semantics of the IP header. In typical networks, virtual machines are fixed and the language used is basic.

With regard to active networks, we can say that the nodes provide a programmable NAPI. If we consider that, in an IP network, the header of the packet provides the input to the virtual machine, we can define an active network as one in which the nodes possess a virtual machine that executes the code contained in the packet headers.

Numerous categories of active networks can be defined on the basis of the following attributes:

– expressive power of the language, which determines the degree with which the network will be able to be programmed. The language can range from simple orders to very highly-evolved languages. The simpler the language, the shorter the computation time. Conversely, the more powerful the language, the greater the degree of customizability that can be implemented;

– possibility of defining a stable state on the basis of the previous messages in the same stream, so as to increase the rate of execution without having to redefine a state of the virtual machine;

– granularity of control, which enables us to modify the behavior of a node for all packets that pass through it, regardless of the stream to which that data packet belongs, or, at the other extreme, to modify the node's behavior only for the particular packet being processed at

the time. All intermediary cases may be found – in particular, common behavior on the same stream or on the same set of streams;

– means of giving programming orders: it is possible to consider that the orders are given to the active nodes by specific packets – e.g. signaling packets – rather than being indicated in a more or less highly-evolved language in the packet header;

– architecture of the nodes, to examine the level of this architecture at which the commands come into play or, in other words, the level at which the programming interface is situated. The architecture can influence the choices of software and hardware. In particular, they can use reconfigurable processors, at higher or lower conceptual levels.

The functions of the nodes of active networks are shared between the runtime environment and the node's operating system. Figure 3.13 illustrates such an architecture of an active network.

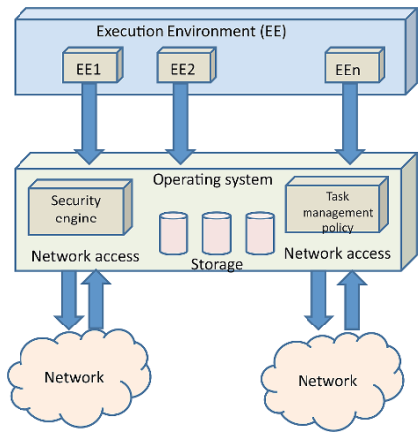



**Figure 3.13.** *Architecture of active networks*

It is possible to send commands to the runtime environment using an encapsulation protocol, called ANEP (Active Network Encapsulation Protocol). The header of an ANEP packet contains a field identifying the type of packet. Several runtime environments may be present in an active node; the address of the node requires an additional address.

The existing interfaces include:

– the interface of access to the runtime environment;

– the interface between the runtime environment and the node's operating system;

– the interface of access to the node's operating system.

## 3.10. Programmable networks

Programmable networks form part of active networks, whose role is to develop a set of software abstractions of the network resources, so that we can access these resources by their abstraction.

The objective of these networks is to render the nodes programmable so as to adapt them to the user demands and the required services. The programming commands, which may be sent either through a signaling network or by user packets containing control programs, can attack the nodes at different levels of abstraction. The IEEE has set up a working group with the purpose of standardizing these interfaces.

## 3.11. Autonomous networks

The concept of an active and programmable network is gradually being supplanted by that of an autonomous network. An autonomous network is a network which does not need a management center or control center to make its decisions. Thus, an autonomous network is a network that can decide for itself how it will behave. This is a concept which was introduced for NGNs (Next-Generation Networks), whose aim is to replace all existing networks by a single IP network, thus integrating all communication media.

An autonomous network must be capable of managing itself, detecting problems, repairing itself and controlling itself when no communication is possible.

The network elements themselves must participate in the construction of an autonomous network with diverse properties, such as resource optimization, context awareness and automatic organization of security. The objective is to understand how to learn which are the right decisions to make, what influence the different elements of the network have and, more generally, how to optimize the network's behavior. The tools to create this type of autonomous system are drawn from multi-agent systems, which we discussed above.

Self-organization of an IP network first requires an overall view of the network and an understanding of the consequences of any event that happens in the network. Then, the network must be capable of reacting to that event.

Autonomous networks can be designed on the basis of networks other than IP for very specific objectives, such as networks with critical missions or interplanetary networks, wherein the control center takes so long to communicate with the probes that it becomes impossible to make decisions in real time.

## 3.12. Autonomic networks

Now we come to autonomic networks which, by definition, are autonomous and spontaneous networks. They are the networks which we defined earlier, but with an added property of spontaneity – i.e. they are real-time systems: the process is capable of reacting autonomously and within an acceptable time-lag.

Figure 3.14 offers an initial illustration of the definition of autonomic networks.

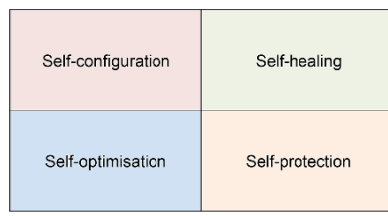



**Figure 3.14.** *Definition of an autonomic network*

Autonomic networks are capable of self-configuring to adapt dynamically to any changes in the environment, of self-optimizing to ensure their operational efficiency is always optimal, self-repairing to ensure it is highly reliable, and self-protecting to ensure the security of the resources and information passing through it.

To perform these different functions, autonomic networks must have a certain number of attributes:

– self-awareness;

– environment awareness;

– self-monitoring;

– self-adjusting.

To achieve these objectives, we need to change the architecture of the networks. Thus, autonomic networks offer a new form of architecture with four planes, where a knowledge plane is added to the usual three: the data plane, the control plane and the management plane. The difference with the SDN architecture stems from the division of the control plane into two sub-planes – the knowledge plane and the control plane itself. The management plane is also an additional plane in this architecture, but it is integrated into the virtualization plane where each software network has its own management plane.

Figure 3.15 illustrates the new architecture of autonomic networks. The purpose of the knowledge plane is to collect all of the knowledge in the network and, for each point in that network, obtain a fairly universal view. The difference with the SDN architecture, here, is that the knowledge is centralized at a single point rather than distributed to each point.

The purpose of the Knowledge Plane is to run the control algorithms found in the Control Plane, which controls the Data Plane, which corresponds to the first four layers of the conventional network architecture. The Management Plane is responsible for the administration of the other three layers.

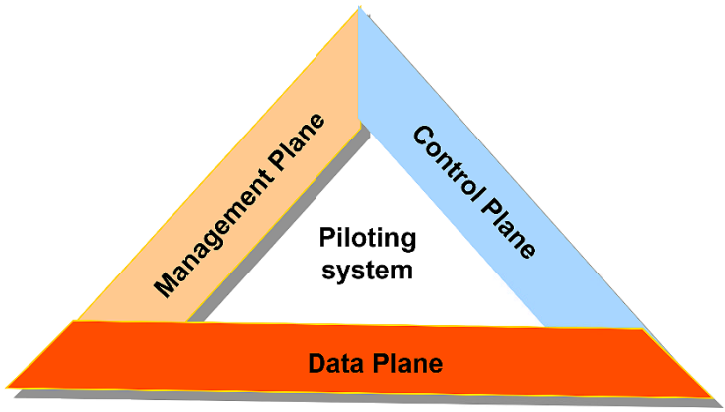



**Figure 3.15.** *The architecture of autonomic networks*

The objective of the knowledge plane is to render the network more intelligent by enabling it to understand its own behavior, which gives rise to a new generation of protocols. Up until now, every control algorithm (routing, quality of service, security, reliability, etc.) has had to go looking for the elements it needed, by itself. For example, a routing algorithm such as OSPF looks for the state of the upstream and downstream links to the input and output points of the network. This information can be exploited by other algorithms, such as an algorithm monitoring QoS, congestion or indeed admission to the network. Using a knowledge plane, this information is found in that plane, and each control algorithm can go in search of it as and when needed.

In time, the standardized protocols should be altered to take account of this knowledge plane. Another advantage of the knowledge plane is the possibility of using information, for a control algorithm, which would not have been able to be taken into account by the normal algorithm.

## 3.13. Situated view

Evidently, it is important not to make the network too cumbersome, so as to be able to transport knowledge of all kinds, and

in vast quantities. With this in mind, there are two possibilities: either to centralize the knowledge, as the SDN architecture does, or to distribute it as widely as possible, making just enough knowledge available, at each point, for decisions to be able to be taken. The distributed solution comes from the situated view technique.

"Situated views" come from the world of artificial intelligence and indicate the integration of pieces of knowledge that are situated in a given view. Thus, we can define a situated view by the number of hops necessary to find the information – e.g. one or two hops, etc.

Figure 3.16 shows a one-hop situated view.

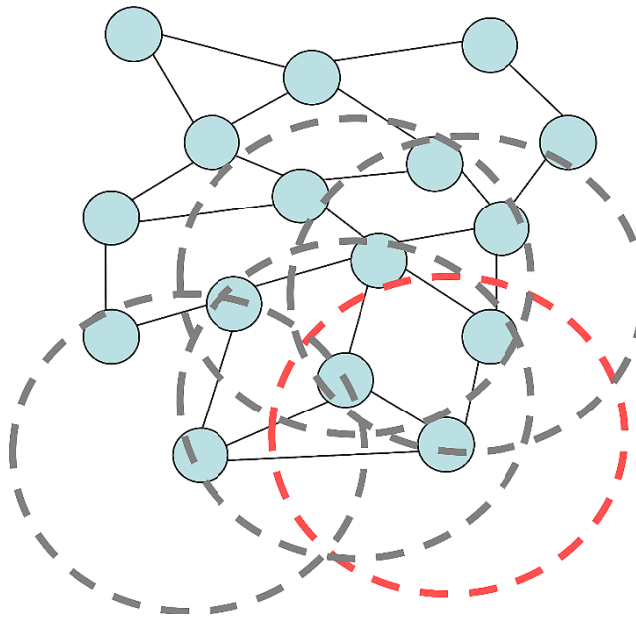


**Figure 3.16.** *One-hop situated view. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

The one-hop situated view exhibits the advantage of being very easily scalable, because the transport of the knowledge is limited. This does not mean that the knowledge cannot diffuse through the network, because a point that receives a piece of knowledge integrates it into its

own knowledge base and forwards it on to its neighbors. This knowledge, however, is less fresh than if the situated view were a little larger.

Thus, it is necessary to optimize the situated view depending on the knowledge needed by the control algorithms. With this in mind, we need to answer the questions: which, where and why? "Which" refers to the knowledge necessary to optimize an algorithm; "where" relates to the span of the situated view; and "when" indicates the refreshes needed for that knowledge to be usable. On the basis of these different parameters, it is possible to generate situated views that are a little more complex than the simple one-hop definition, including the information of links and all the seconds.

Until 2008, networks were remarkably static, and a network engineer often needed to be present to deal with things when a problem arose. In the case of very large networks, tens of network engineers could be needed to perform maintenance and deal with the diverse problems that emerged.

The objective of self-adjusting networks is to offer automated piloting of the network by a program capable of managing the control algorithms in a coordinated manner, and thereby optimize the network's operation.

At the start of the 2000s, an initial attempt was made, consisting of using programmable networks and active networks. The research was not entirely conclusive, for reasons of security and cost. A new generation was launched in 2005, with the autonomic networks we have just presented.

## 3.14. Conclusion

The Cloud requires centralization, with its huge datacenters capable of handling enormous masses of data and of computing forwarding tables for a very high number of data streams. Unfortunately, this solution is not entirely satisfactory, because the reaction times are long, and reliability and security pose numerous

problems, which considerably increase as the networks become more complex. This is the reason for the advent of distributed solutions using far smaller datacenters, much nearer to the users. However, this type of architecture requires a greater degree of intelligence to coordinate the set of controllers.

Slowly but surely, intelligence is being included in networks. This intelligence covers communication, reasoning and decision-making. Multi-agent systems provide the mainstay of this intelligence, which is able to make control- or management decisions when required. We are only at the very beginning of the story, but intelligence has already been omnipresent for a number of years at time of writing.

This intelligence can be exercised from huge interconnected datacenters. In this case, it is centralized for the management and control of a sub-network. The interconnection of those sub-networks requires a greater degree of distribution. When the networks are controlled by datacenters of all sizes, the intelligence will need to be very widely distributed.

Whilst intelligence serves to control the networks, it also has other impacts. In particular, security is one of the main beneficiaries of this intelligence. For example, an intelligent component, such as a neural network, is capable of analyzing what a person enters on a keyboard and shutting down communication if that input is not recognized.

Intelligence in networks, therefore, is a vast field for R&D, which should lead to the implementation of a much greater intelligence in all domains relating to telecommunications. Another underlying vision is that of the reduction of the costs in terms of specialized personnel for surveillance, maintenance and, more generally, all the operations necessary for the survival of a network. It should become possible to reduce the number of engineers and technicians in the networking field by 50-80%, and thereby make substantial savings. However, it should be noted that new professions will emerge, with architects and high-level engineers associated with the domain of networks.

# 4

# New-generation Protocols

Protocols are currently undergoing numerous mutations, owing to the advent of the Cloud and of SDN architectures. The legacy protocols based on routing of IP packets and MPLS (MultiProtocol Label Switching) are still in place for the connection of users or data transport, offered by the major operator networks. However, an entirely new generation of networks is emerging and fighting to earn a place in the new arsenal of operators and Cloud providers. Firstly, there are the networks within datacenters, which must provide a very fast link between the servers, of which there are thousands, or even tens or hundreds of thousands. The protocol cited most at present is TRILL (Transparent Interconnection of Lots of Links). TRILL is a level-2 Ethernet network with extensions to facilitate routing in a level-2 Ethernet environment. Very broadly speaking, we can qualify TRILL as an Ethernet network encapsulated within an Ethernet network, enabling us to have a hierarchy of addresses and to carry out routing. We shall come back to this later on.

Then, there are inter-datacenter networks, which also carry enormous quantities of data, going from one sub-network to another sub-network. There are a certain number of possibilities that are mentioned in the literature and are found in reality with Cloud management providers. In general, they are extensions of VLAN (Virtual LAN) techniques to maintain consistent VLAN numbers, even when the datacenters are not in the same sub-networks. In addition, the number of VLANs is limited by numbering zone, which

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

only allows for 4096 VLANs, because of the 12 bits in that zone. This category of protocols includes:

– VXLAN (Ethernet in UDP), originally developed by VMware;

– NVGRE (Ethernet in IP), developed by Microsoft;

– 11aq (EVLAN), stemming from the standardization of Carrier-Grade Ethernet.

At the very least, the following two protocols need to be added to this list:

– MPLS, developed by operators and standardized by the ITU and the IETF,

– LISP (IP in IP) supported by numerous hardware manufacturers.

The last of these protocols – LISP (Locator/Identifier Separation Protocol) – is fundamentally different from the previous ones, because it is a level-3 protocol, which is the encapsulation of an IP packet within an IP packet. The reason for this encapsulation is the need to have two IP addresses associated with an end device: the inner address to identify the intended receiver and the outer address for routing. This latter address denotes the physical position of the receiver. This solution means that the virtual machines can be moved but retain the same identity by modifying the IP address of the placement of the virtual machine. We shall also come back to this in greater detail later on.

Finally, the rising protocol corresponds to that developed for SDN: it is OpenFlow, which was introduced by Nicira, later acquired by VMware. We shall begin with a look at this protocol.

Remember that, as illustrated by Figure 4.1, SDN (Software-Defined Networking) is the solution which decouples the data plane from the control plane. The computations and control algorithms no longer take place directly in the nodes of the network, routers or switches, but in a server situated somewhere in the network, called the controller. The controller may be a physical device or a virtual machine situated in a datacenter of greater or lesser size.

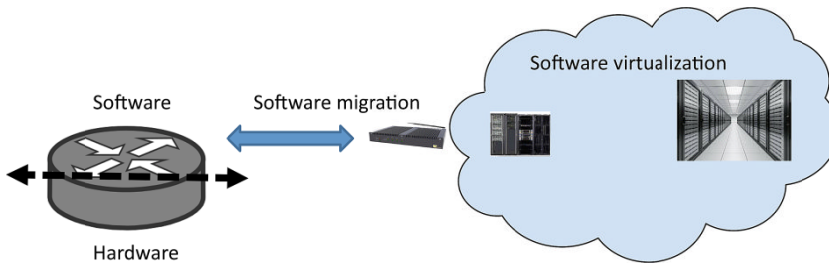



**Figure 4.1.** *SDN architecture*

The data plane includes the forwarding of packets. This solution ensures the hardware is independent of the software networks and that numerous parameters are taken into account. Indeed, this decoupling enables the controller, which is usually centralized, to recover precise information from each client in the network and to perform complicated computations using load and QoS algorithms, so that each flow in the network has its own forwarding tables in the network nodes. In the extreme case scenario, each flow could have its own tables, taking account of all the characteristics of the same client: performance, energy consumption, reliability, security, resilience, cost, etc. This solution cannot be currently scaled, but it could become scalable in the future with major decentralization and high-quality inter-controller protocols (see the eastbound and westbound interfaces in the SDN architecture discussed in Chapter 2). Let us begin by detailing the OpenFlow protocol.

## 4.1. OpenFlow

Figure 4.2 shows an OpenFlow switch and an OpenFlow controller. The term "switch" is used for the OpenFlow nodes because, as we shall see, paths are usually determined by the controller.

The OpenFlow switch has two parts: the part that contains the queues, the frame transmitters and frame receivers, with the flow tables associated therewith, and the second part which governs

communication with the controller using the OpenFlow signaling protocol.

The first part contains all the necessary elements for the physical transport of the frames into the node and the tables used to direct the flows into the right egress queue. There may be a table for each flow such as a table for a set of flows multiplexed on the same path. OpenFlow provides the necessary elements from the controller to create, manage and destroy flow tables. The second part relates to the communication between the node and the controller, and the information that must be sent between them.

OpenFlow uses a secure SSL/TLS channel to authenticate both ends of the communication, which greatly reduces the risk of attack on the communication under way. OpenFlow offers the means of identifying the flows of packets using level 1, 2, 3 and 4 data. OpenFlow also transports actions to indicate what needs to be done to the flow, and it feeds back very precise statistics to the controller so that the path determination algorithm can do its job with near-perfect knowledge of the state of the network. As the controller is centralized, there is a slight time lag in the transmission of the data.

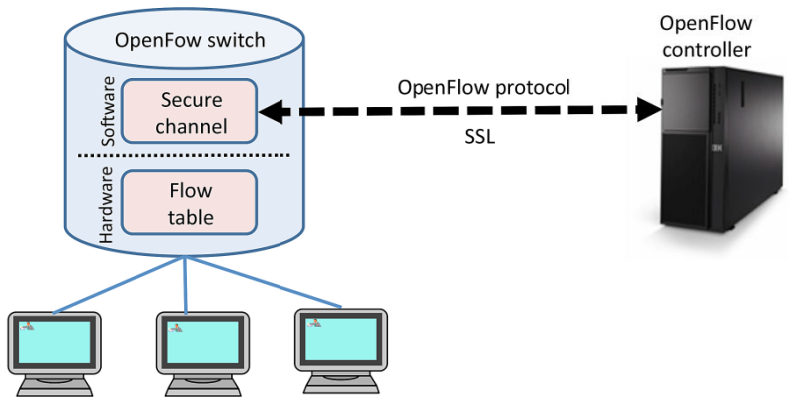



**Figure 4.2.** *OpenFlow protocol*

Figure 4.3 shows a more general OpenFlow network. The controller is on the right of the figure, and opens SSL/TLS connections

with all the nodes in the network. Potentially, certain nodes may not be OpenFlow-compatible, and form a sub-network carrying OpenFlow signaling. In general, the nodes are compatible with OpenFlow, meaning that they are able to interpret OpenFlow commands and introduce actions on the flows in the flow tables or transform them to conform to forwarding tables or switching tables. The packet flows never go through the controller, except the case where a function, such as DPI (Deep Packet Inspection), is implemented in the controller, to analyze the set of all packets in that flow. In Figure 4.3, no flow goes through the controller, except the first packets, which are needed to determine the demands and the type of flow being dealt with.

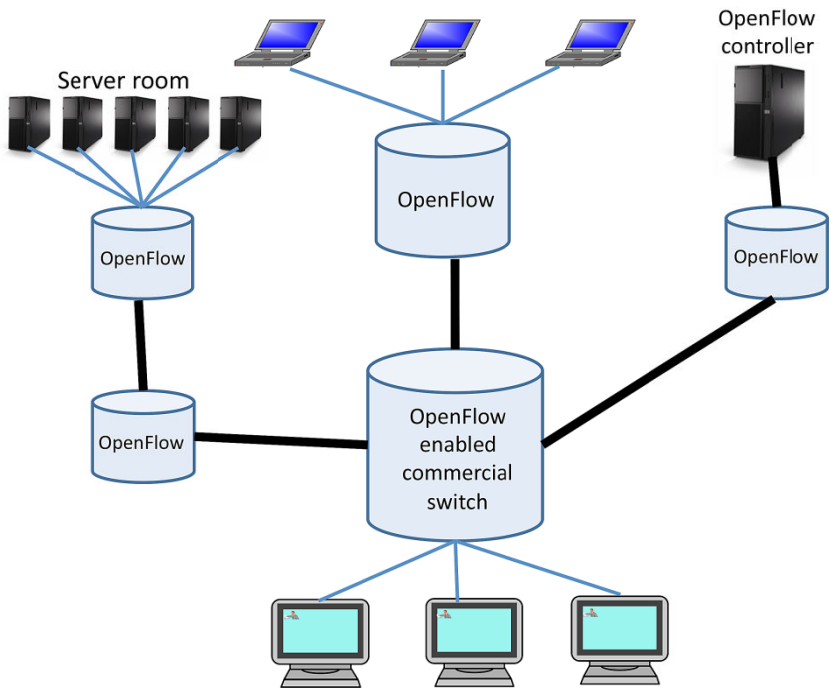



**Figure 4.3.** *OpenFlow protocol in a network*

The fields in OpenFlow are shown in Figure 4.4. This figure describes the flow table, which contains the three types of information listed above. The first field enables matching between a row in the

table and the flow. The flow is described by the ingress port number and the Ethernet addresses, VLAN number, VLAN priority, IP addresses, protocol and type of service fields, and finally, the port numbers of the TCP or UDP transport layer.

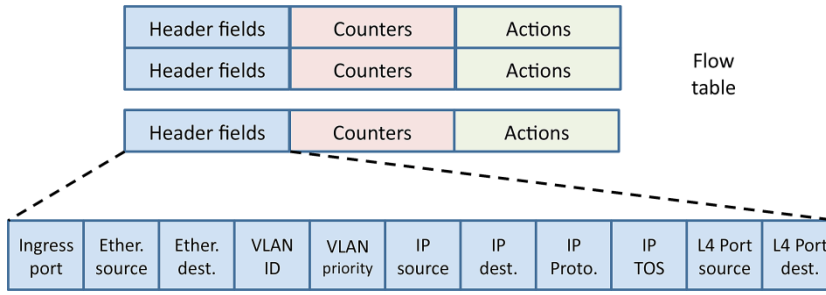



**Figure 4.4.** *Fields in the OpenFlow protocol*

The flow table is supplemented by values of the frame counters, byte counters, etc., giving a precise indication of the statistics of flows across all ports in the network. Therefore, the controller has a very precise and complete view of the whole network under its management.

Amongst the numerous possible actions which can be transported by OpenFlow signaling, the most common include:

– sending a packet over a list of ports;

– adding/rejecting/modifying a VLAN Tag;

– destroying a packet;

– sending a packet to the controller.

The first generation OpenFlow was standardized by the ONF (Open Network Foundation) in 2009. Since then, new versions have been released, essentially containing extensions, such as:

– version v1.1 in 2011, to take account of the MPLS and Carrier-Grade Ethernet Q-in-Q techniques, which we shall examine a little

later on. Multicasting and management are also taken into account, as is the possibility of having multiple routing tables;

– version v1.2 introduces the possibility of controlling networks using the protocol IPv6. This version also goes into much greater detail about the statistics of flows, and introduces multiple controllers when the network can be split into several parts, with each one being handled by a separate controller;

– the next version, v1.3, extends the range of protocols taken into account still further, including MAC-in-MAC from Carrier-Grade Ethernet and the possibility of having multiple communication channels between a controller and each of the OpenFlow switches in the network;

– version v1.4 takes a new direction, with optical network control.

These different versions of the OpenFlow protocol are described in Figure 4.5. Version 1.5 is due to be released in 2015. There could even be a version 2.0, because there are a great many working groups that have proposed extensions to include most of the main types of networks around today.

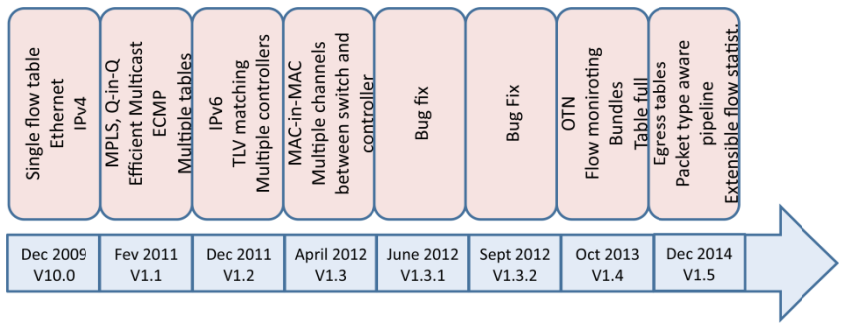



**Figure 4.5.** *The different ONF standards pertaining to the OpenFlow protocol*

Many open-source programs are OpenFlow-native or OpenFlow-compatible – to mention just a few:

– Indigo: an open-source implementation which is executed on a physical machine and uses the characteristics of an ASIC to execute OpenFlow;

– LINC: an open-source implementation that runs on Linux, Solaris, Windows, MacOS and FreeBSD;

– Pantou: OpenFlow for a wireless environment, OpenWRT;

– Of13softswitch: a software switch produced by Ericsson;

– XORPlus: an open-source software switch;

– Open vSwitch: an open-source software switch developed by Nicira and integrated into the Linux kernel (in versions 3.3 and later). Open vSwitch is greatly used by numerous manufacturers in their architecture.

Similarly, many switches and routers are OpenFlow-compatible. This long list includes the following:

– NOX: NOX was the first OpenFlow controller;

– FlowVisor: a Java OpenFlow controller that behaves like a transparent proxy between an OpenFlow switch and multiple OpenFlow controllers;

– POX: a Python-oriented OpenFlow controller with a high-level SDN interface;

– Floodlight: a Java OpenFlow controller;

– OpenDaylight: OpenDaylight is an open-source project for a modulable platform which contains a controller at its center. The OpenDaylight controller is used as a primary controller by numerous manufacturers, even when those manufacturers have other proprietary solutions that they could use.

In terms of the controllers, in addition to those mentioned above, there are numerous developments under way, compatible with OpenFlow, but handling many other northbound and southbound interfaces that have been developed to satisfy manufacturers' needs. Certainly the best known of these is OpenDaylight, which we mentioned in the list above. The architecture is represented in

Figure 4.6. The version illustrated here is Helium – the successor of the Hydrogen version. In this version, we see some of the modules of the controller and the northbound and southbound communication interfaces. This controller is OpenFlow-compatible, but many other possibilities are also acceptable, including OvSDB, NetConf, SNMP, etc.

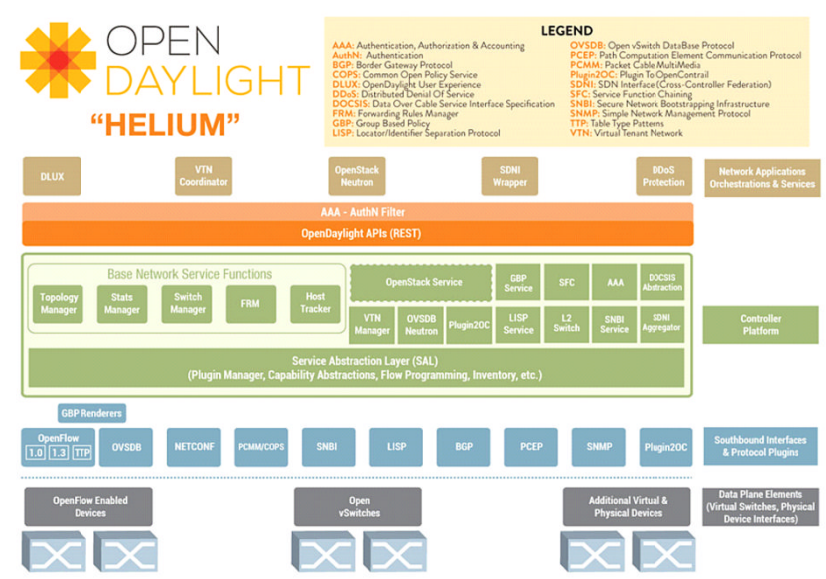



**Figure 4.6.** *OpenDaylight controller*

We find the same type of properties for similar controllers such as OpenContrail and Floodlight.

OpenFlow is undoubtedly the best known of the protocols used for the southbound interface, but nothing is yet finalized in the advance of SDN. This signaling protocol has the advantage of being relatively simple and well suited for its intended use.

Let us now examine a certain number of protocols for the Cloud, and more specifically for interconnecting datacenters. For this

purpose, it is common to use VLANs, but with extensions so as not to have a significant limitation on the number of VLANs supported. One of the most widely-used techniques is a direct extension of VLANs: VXLANs.

## 4.2. VXLAN

VXLAN (Virtual eXtensible LAN) is a solution capable of delivering communications in Clouds between datacenters. The technology is fairly similar to that used for VLANs and Carrier-Grade Ethernet extensions. It was originally developed by Cisco and VMware. The drawback to the basic solution of VLAN IEEE 802.1Q is the limitation to 4096 VLANs. VXLAN enables us to extend the basic technology in parallel to Carrier-Grade Ethernet.

In order to move from one sub-network to another, in VXLANs we consider that we can go through any network, and therefore we need to go back through the transport layer. For this reason, the Ethernet frame to be transported between datacenters is encapsulated in a UDP message which, itself, is encapsulated in an IP packet and then in an Ethernet frame. These successive encapsulations are shown in Figure 4.7.

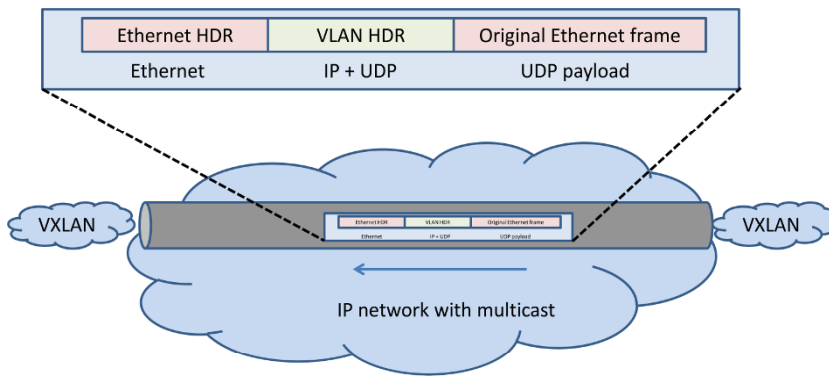



**Figure 4.7.** *VxLAN protocol*

To the basic Ethernet frame, we add a VXLAN identification zone of 24 bits, meaning we can achieve over 16 million VLANs and then a 64-bit UDP field, which is then encapsulated in an Ethernet frame with fields, source address, destination and VLAN, associated with VXLAN. The basic Ethernet frame therefore forms the "data" part of the UDP message, which, itself, is transported in an Ethernet frame. This property enables us to create VLANs beyond an Ethernet domain. These encapsulations are represented in Figure 4.8.

It should be noted that the overload is significant because the VXLAN frame adds 36 bytes to the basic frame.

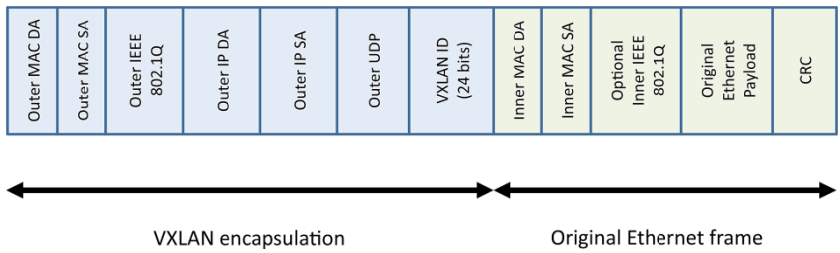



**Figure 4.8.** *VXLAN encapsulation*

## 4.3. NVGRE (Network Virtualization using Generic Routing Encapsulation)

Another protocol which is also being driven forward by the IETF is NVGRE (Network Virtualization using Generic Routing Encapsulation). It is supported by various industrial players, including Microsoft. This protocol, like the former, enables us to pass through an intermediary network between two datacenters, using an IP network. In order to preserve the value of the VLAN, we need to encapsulate the basic Ethernet frame in an IP packet, itself encapsulated in frames, to pass through the IP network. The original Ethernet frame is retrieved on the other side of the network. Figure 4.9 shows the tunnel which is opened in the IP network to transport the original frame.

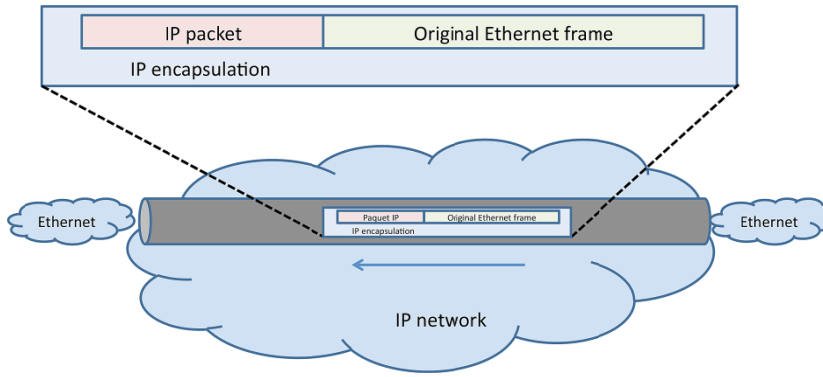



**Figure 4.9.** *NVGRE protocol*

To interconnect datacenters, the classic solutions such as MEF (Metro Ethernet Forum) Ethernet, GVLANs (extended VLANs) or indeed Carrier-Grade Ethernet offer entirely usable solutions. We shall begin by looking at the MEF Ethernet solution.

## 4.4. MEF Ethernet

MEF (Metropolitan Ethernet Forum) networks have been on the scene for quite some time. Originally, their purpose was to interconnect company networks in a single, very high capacity hub. However, they are also perfectly appropriate for interconnecting datacenters. MEF networks use switched Ethernet networks at 1, 10, 40 and 100 Gbps.

To achieve telephony over IP and services requiring strict time constraints, Ethernet operator networks (Carrier Grade Ethernet) enable us to introduce priorities, with the difference that the field IEEE 802.1p, which serves for the introduction of these priorities, has only 3 bits. These 3 bits can only encode 8 different levels of priority, as opposed to the 14 levels defined by the IETF for DiffServ services.

The levels of priority proposed by the MEF are as follows:

– 802.1p-6 DiffServ Expedited Forwarding;

– 802.1p-5/4/3 DiffServ Assured Forwarding;

– 802.1p-5, which presents the lowest loss;

– 802.1p-3, which presents the highest loss;

– 802.1p-2 DiffServ Best Effort.

In the Ethernet environment, flow control is generally a tricky problem. Various proposals have been made to improve it. In particular, back-pressure methods involve the sending of control messages by overloaded switches, thus allowing the connected switches to cease their transmissions to the congested node for a period of time indicated in the control guidelines.

The choice made by MEF is based on the DiffServ control scheme, where we find exactly the same parameters:

– CIR (Committed Information Rate);

– CBS (Committed Burst Size);

– PIR (Peak Information Rate);

– MBS (Maximum Burst Size).

This solution demonstrates the ubiquitousness of the world of Ethernet, which offers a large number of solutions capable of competing amongst themselves to satisfy the new generation necessary in the world of cloudified networks.

## 4.5. Carrier-Grade Ethernet

Ethernet was designed for computer applications, rather than for applications in the world of telecommunications, which requires particular qualities which we call “carrier grade”. In order to conform to the requirements of the operators, therefore, the Ethernet environment has had to adapt. We now speak of Carrier-Grade Ethernet – i.e. a solution acceptable for telecom operators with the control- and management tools necessary in this case. This mutation essentially relates to switched Ethernet.

Carrier-Grade Ethernet must possess functions that are found in telecommunications networks – most notably the following:

– reliability, which means that there must be very few failures. The MTBF (Mean Time Between Failures) needs to be at least 50,000 hours;

– availability, which must achieve the classic values in telephony – i.e. a functioning state 99.999% of the time. This value is far from being achieved by conventional Ethernet networks, which actually only offer availability around 99.9% of the time;

– protection and restoration. When a failure occurs, the system must be able to recover within a maximum of 50 ms. This duration comes from telephony, where breakdowns lasting longer than this are unacceptable. SONET networks, for example, reach this value for their reconfiguration time. The solutions generally use redundancy, either total or partial, which involves engaging another path, arranged in advance;

– performance optimization by active or passive monitoring. The operations are not all entirely homogeneous when the packet flows vary. Thus, we need to adapt the flows so they can flow without a problem;

– the network must be able to accept the SLAs (Service Level Agreements). The SLA is a typical notion in an operator network when a customer wants to negotiate a service guarantee. The SLA is determined by a technical part – the SLS (Service Level Specification) – and an administrative part whereby penalties are negotiated if the system does not satisfy the requirements;

– management is also an important function of operator networks. In particular, failure-detection systems and signaling protocols must be available in order for the network to work.

From a technical standpoint, Carrier-Grade Ethernet is an extension of VLAN technology. A VLAN is a local-area network in

which the machines may be a very great distance apart. The objective is to make this network operate as though all of the points were geographically close to one another to form a local-area network. A VLAN may contain several users. Each Ethernet frame is diffused to all the machines in the VLAN. The tables which determine the frame switching are fixed and may be viewed as switching tables in which the addresses of the recipients are references.

When the VLAN has only two points, the VLAN defines a path. This is the vision which was employed for Carrier-Grade Ethernet. Paths are formed by determining VLANs. The path is unique and simple if the VLAN has only two points. VLAN introduces a multipoint if there are more than two points.

The problem with this solution stems from the limited size of the VLAN field, which is only defined by 12 bits. This is perfectly appropriate in the context of a company network with standard Ethernet switching, but becomes completely insufficient for Carrier-Grade Ethernet, which is aimed at operator networks. Therefore, the size of the VLAN field has had to be increased.

Carrier-Grade Ethernet can be subdivided into various solutions of extension of the VLAN zone, all of which are illustrated in Figure 4.10. The most typical solution consists of using the IEEE 802.1ad standard, which has a variety of names: PB (Provider Bridge) Ethernet, QiQ (Q in Q) or cascading VLAN. IEEE 802.1ah is also known as MiM (MAC-in-MAC) or PBB (Provider Backbone Bridge). The most advanced solution is called PBT (Provider Backbone Transport), or pseudo wire (PW) over PBT. A PseudoWire is an MPLS-based tunnel whereby Ethernet frames can be transported over an IP network with QoS guarantees.

The solutions described in the foregoing sections are illustrated in Figure 4.10.

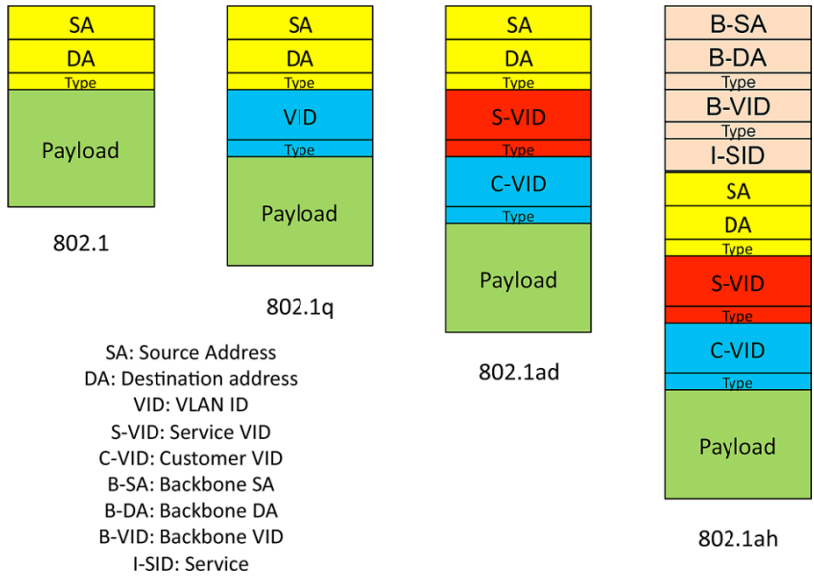



**Figure 4.10.** *The different versions of Carrier-Grade Ethernet. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

Let us first examine PB (Provider Bridge) Ethernet technology. The ISP adds a VLAN number to that of the customer. Thus, there are two VLAN numbers: the C-VID (Customer-VLAN ID) and the S-VID (Service-VLAN ID). The introduction of the ISP enables us to extend the concept of the VLAN to the operator's network without destroying the user's VLAN. This solution enables us to define the transmission to be carried out in the operator's network. As the field of the Ethernet frame, where the VLAN ID is indicated, has a length of 12 bits, this enables us to define up to 4094 entities in the operator network. These entities may be services, tunnels or domains. However, whilst 4094 is a sufficient value for a company, it falls far short of an operator's needs. Implementations exploit a reference translation to extend the domain, but this increases the complexity of managing the whole network. Thus, this solution is not appropriate for large-scale networks.

The solution proposed by the group IEEE 802.1ah PBB (Provider Backbone Bridge) improves on the previous one by switching the

frame traffic on the MAC address. This solution, said to be MIM (MAC-in-MAC), encapsulates the client's MAC address in an operator MAC. This means that the core operator only needs to know its own MAC addresses. In the PBB network, the matching of the user MAC addresses and network MAC is known only by the edge nodes, thus preventing the combinatorial explosion of the number of MAC addresses.

The third solution, called PBT (Provider Backbone Transport), is fairly similar to the MPLS technique, but provides the properties necessary for Carrier Grade, such as unavailability of less than 50 ms. In a manner of speaking, this is a secured MPLS tunnel. The PBT tunnel is created like an MPLS tunnel, with the references corresponding to the endpoints of the network. The client and server VLAN numbers are encapsulated in the MPLS tunnel, which may, itself, have differentiation into operator VLANs. Thus, the real reference is 24 + 48 bits, so 72 bits.

The last solution is that of the service PS (PseudoWire) offered by MPLS. In this case, the user and operator VLANs are encapsulated in an MPLS service tunnel, which may, itself, be encapsulated in an MPLS transport tunnel. This solution stems from the encapsulation of tunnels in MPLS.

Carrier-Grade Ethernet technology is of interest to numerous operators. Only time will tell which solution ultimately wins out. However, even at this stage, it can be conclusively stated that VLAN-within-VLAN encapsulation will be present in all such solutions.

## 4.6. TRILL (Transparent Interconnection of a Lot of Links)

TRILL (Transparent Interconnection of Lots of Links) is an IETF standard implemented by nodes called RBridges (routing bridges) or TRILL switches. TRILL combines the strong points of bridges and routers. Indeed, TRILL determines a level-2 routing using the state of the links. RBridges are compatible with level-2 bridges defined in the IEEE 802.1 standard, and could gradually come to replace them. RBridges are also compatible with IPv4 and IPv6 routers, and

therefore perfectly compatible with the IP routers used today. Routing done with the protocol IS-IS at level 2 between the RBridges replaces STP (Spanning Tree Protocol) in TRILL. Figure 4.11 shows the processes employed by TRILL.

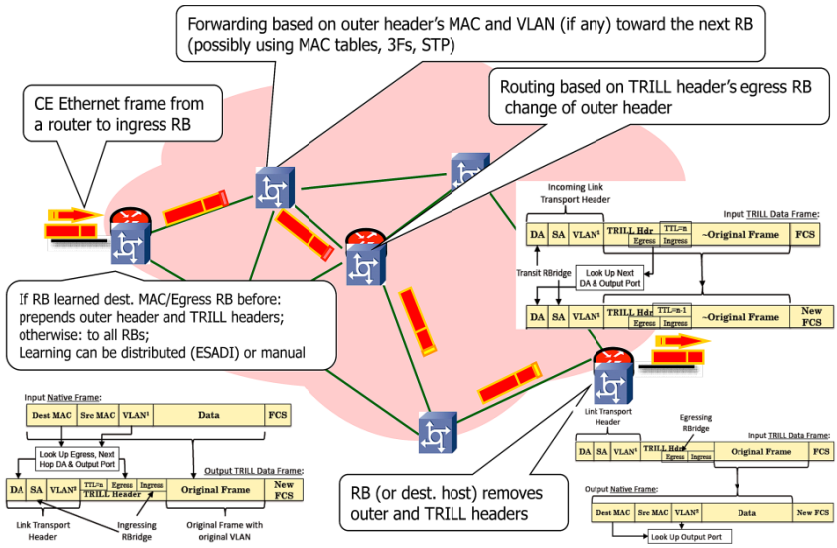



**Figure 4.11.** *TRILL protocol*

Using level-2 routing between the Ethernet addresses means that we no longer have to configure level 3 and the associated IP addresses. The link state protocol used for routing may contain additional TLV (type, length, value) data. In order to prevent potential loops when rerouting, RBridges maintain a "hop count" – i.e. they count the number of nodes through which the message is passed. When this number reaches a maximum value, determined by the operator, the frame is destroyed.

When an Ethernet frame reaches the first RBridge in the network, called the IRB (Ingress RBridge), an extra header is added: the TRILL header. It is decapsulated by the ERB (Egress RBridge). In the TRILL part of the new frame, we find the address of the ERB, which makes this a routing technology, as it uses the recipient's address rather than a reference. This address of the RBridges pertains to 2 bytes, as

opposed to the convention 6 bytes occupied by the Ethernet frame. Thus, addressing is done on the basis of addresses defined by the user him/herself. The added header contains 6 bytes in total, terminating with the ingress and egress addresses, preceded by the hop count and a flag.

The Ethernet frames are recovered after decapsulation at their egress. Thus, we see a classic mode of forwarding – either using the VLAN number, with the Ethernet address serving as a reference, or decapsulating the Ethernet frame to retrieve the IP packet.

An interesting point is the possibility to transfer the frames from the same flow using several paths simultaneously: this is known as multipath, or ECMP (Equal Cost MultiPath), which enables us to detect the different paths of equal cost and point the frames along these various paths.

## 4.7. LISP (Locator/Identifier Separation Protocols)

LISP (Locator/Identifier Separation Protocol) was developed to facilitate the transport of virtual machines from one datacenter to another without changing the IP address of the virtual machine. In order for this to work, we need to separate the two interpretations of the IP address: the identifier of the user machine and the locator used for routing. If we wish to preserve the address of the virtual machine, it is necessary to differentiate these two values. This is what LISP does, but it is not the only protocol to do so: HIP (Host Identity Protocol) and SHIM6 (Level 3 Multihoming Shim Protocol for IPv6) also differentiate the two interpretations, but with mechanisms based on the destination machines.

With LISP, the routers take care of the association between the address of the destination machine – the EID (Endpoint ID) – and the address used for routing – the RLOC (Routing-Locator). When a communication is begun between an addressee machine and a server machine, the traffic is intercepted by an “iTR” (ingress Tunnel Router), which must determine the appropriate RLOC to access the server machine, whose address is that of the EID. The network egress

to reach the EID is handled by the eTR (egress Tunnel Router). To perform this operation, the router consults a directory service – the EID-RLOC. Once it has obtained the receiver's RLOC, the packet can be routed to that receiver. This process is not visible from the machine and the server.

Figure 4.12 illustrates the main elements of LISP.

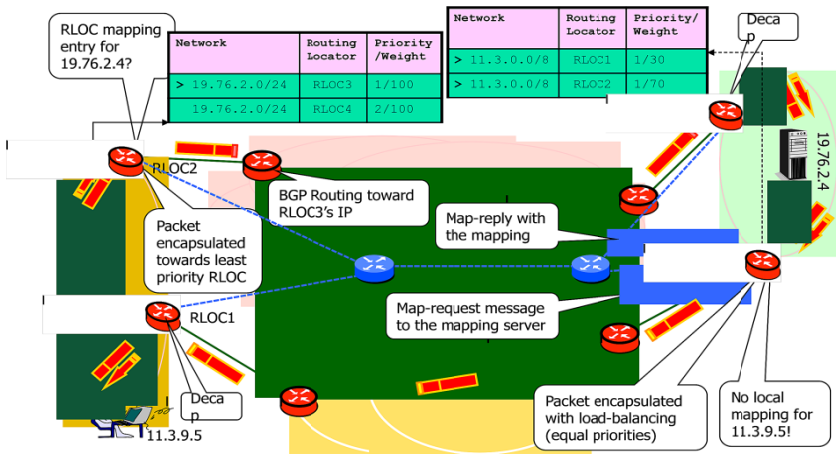



**Figure 4.12.** *LISP protocol*

It should be noted that LISP enables us to use addresses other than those of level 3 in IPv4 or IPv6, such as a GPS location or a MAC address.

## 4.8. Conclusion

The networks found in Cloud architectures are essentially Ethernet-based, and generally use a level-2 architecture. In other words, less use is made of IP, which is far more complex and a good deal slower, with the exception of LISP, which is favored by numerous manufacturers and operators because of its appropriateness for the Cloud environment.

Thanks to the considerable success of level 2, working group 802 at the IEEE is increasing the number of their campaigns in other directions, such as wireless networks, for which wireless Ethernet has become the main standard, access networks or home networks. In Chapter 6, we shall take a look at the new standards that are emerging in this domain.

# 5

# Mobile Cloud Networking and Mobility Control

Mobile Cloud Networking (MCN) is a solution which has become very important in the context of mobile networks. The basic element is the mobile terminal, which moves around and requests services from a Cloud which, for its part, is fixed. Mobile Cloud also refers to technologies where the Cloud itself is mobile. It moves around with the client. The mobile Cloud may be virtual but also physical. We shall begin, in the first section, by examining MCN, and then in the second section we shall look at mobile Clouds, before concluding this chapter with a discussion of the means of control of mobility of the terminals and users.

## 5.1. Mobile Cloud Networking

The exact definition of "Mobile Cloud Networking" is very controversial, because there is no real definition and no clear architecture. There are two predominant orientations in the field of MCN:

– an application orientation, which means that a mobile device with limited resources is able to handle applications which require intensive computations or more memory than the terminal has;

– a network orientation, which involves the optimization of algorithms for the control of mobile services.

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

In the context of application orientation, we can cite various applications which correspond to this framework, such as Rich Mobile applications, "Mobile Cloud gaming" or indeed MGaaS (Mobile Game as a Service), augmented reality or mobile applications involving intensive computations such as OCR (Optical Character Recognition) or natural language use.

In the context of network orientation, we might find, for example, firewalls for mobile devices, handover control or the management of mobile terminal attachment.

The architectures for "Mobile Cloud Networking" cover the different classifications mentioned above: Cloud networking for mobile terminals, local Clouds, virtual Clouds or indeed Cloudlets. The first architecture is illustrated in Figure 5.1.

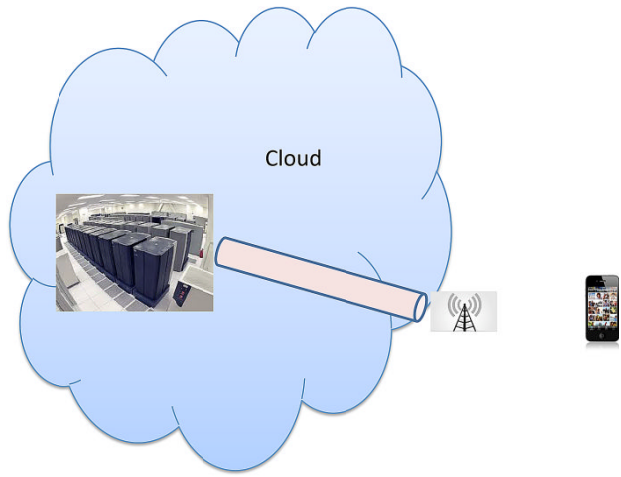



**Figure 5.1.** *An architecture for Mobile Cloud Networking*

This architecture shows a mobile terminal – a smartphone – which uses a central Cloud to handle a certain number of applications. These applications are CPU intensive, a very high memory requirement or the use of specific resources which cannot be accommodated on the smartphone as a "big data" depository. The communications between the smartphone and the Cloud must be reasonable so that the wireless

interface can handle them without any problems. In summary, the terminal is a lightweight machine which calls on the Cloud to carry out hungry processes.

Figure 5.2 illustrates the second architecture, which is fairly similar to the previous one but where the Cloud is no longer central, but is local, or at least not too far from the user. The resources can be hosted on other mobile devices that are connected to the Cloud. The mobile devices may themselves form a Cloud. In other words, neighboring mobile terminals, if they are inactive, can perfectly well serve to perform a computation, storage, or even a network in that environment. We shall see these scenarios for Mobile Cloud at the end of this chapter.

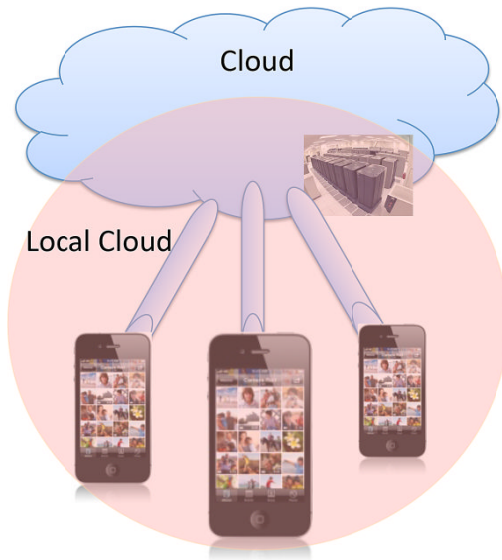



**Figure 5.2.** *An architecture for local Mobile Cloud Networking*

This "Mobile Cloud" is built from connected machines and possibly a local datacenter, which may be very local with micro-datacenters or nano-datacenters or pico-datacenters or even femto-datacenters. Overall, the small datacenter can be situated at the level of the DSLAM of the access router or even the Home Gateway.

Figure 5.3 presents a third case of MCN with a virtual Cloud. In this solution, the client with his/her mobile device wishes to connect to a whole set of Clouds to cover all the services he/she requires. As a single Cloud is usually insufficient to contain all the services necessary for a user, the user needs to request connection to several Cloud providers to obtain the services he/she needs. As it may be tricky and complex to know all the solutions, one possibility is to call on an intermediary which has extensive knowledge of all the Clouds. This intermediary is able, at any given time, to choose the best possible Cloud provider for a given application. The Cloud is virtual because it is represented by an intermediary virtual machine capable of connecting the client to the best Cloud to deliver the required application. Another name given to this solution is "Sky", and the intermediary is a Sky provider. One of the difficulties for the Sky provider is intermediation with Clouds which may be very different to one another, but which must not appear any differently to the end user.

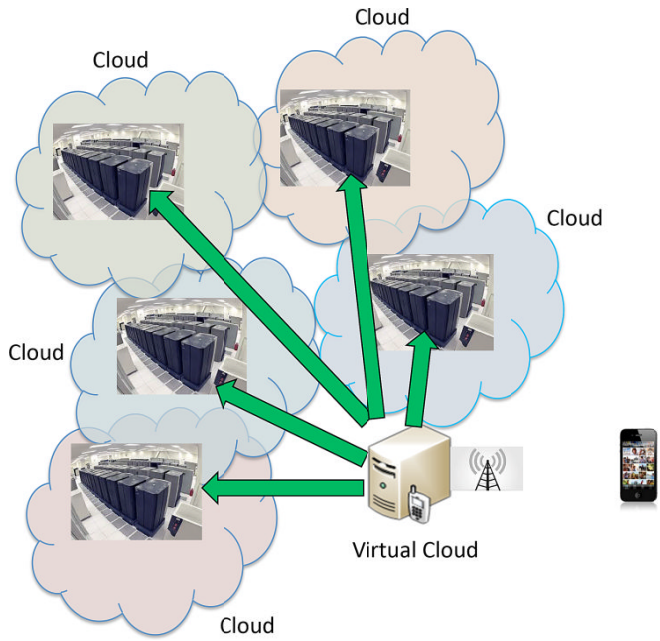



**Figure 5.3.** *A third Mobile Cloud Networking architecture*

Finally, the fourth architecture encountered under the auspices of Mobile Cloud Networking also pertains to the use of a small Cloud – a Cloudlet – which moves with the client. In fact, it is not that the Cloud actually moves, but the connection to different Cloudlets gives the impression that the Cloud is moving with the client. With each handover, the mobile terminal attaches to a new Cloudlet, and the client's virtual machines migrate over to the new small datacenter. This solution is illustrated by Figure 5.4.

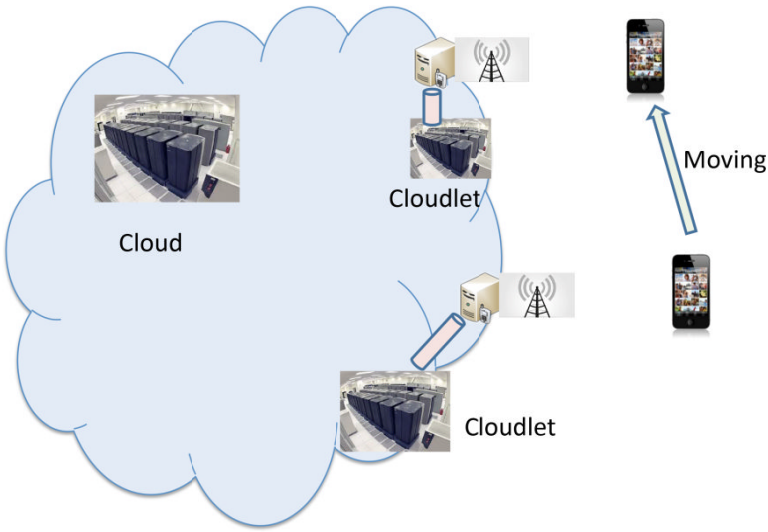



**Figure 5.4.** *A fourth architecture for Mobile Cloud Networking*

A Cloudlet is a concept of a small Cloud situated in a zone with very high demand. The Mobile Cloud provider assigns the most appropriate Cloudlet to respond quickly to the service request of the moving client.

In summary, Mobile Cloud Networking architectures are based on a hierarchy of Clouds to optimize different criteria:

– performance and reliability of applications for mobiles;

– performance of control applications for mobiles;

– minimization of energy consumption by the terminals, datacenters, etc.;

– availability;

– high security:

- m-commerce,

- m-Cloud access,

- m-payment.

The architectural differences we have described can be combined to optimize the above criteria. Once again, we see the hierarchy of datacenters ranging from enormous datacenters to the minuscule "femto-datacenters". It is not easy to optimize these environments, even though the architecture is sometimes relatively well defined. Often, it appears that the system of the future is Clouds that form easily and are just as easily transformed, and is beginning to gain momentum, known by the name "mobile Cloud", which we shall now examine.

## 5.2. Mobile Clouds

A "mobile Cloud" is a set of small datacenters which, once they are connected, form a Cloud. The difficulty lies in the mobility of such Cloudlets and the diversity of forms of mobility, so the Cloudlets can attach and detach. If all the Cloudlets or mini-datacenters move simultaneously, then it is a VANET (Vehicular Area Network) which supports the mobile Cloud. On the other hand, if the mobiles which transport the mini-datacenters move independently of one another, with no coordination between them, the mobile Cloud has greater difficulty in forming and evolving on the basis of the movements. Figure 5.5 shows an example of a mobile Cloud, wherein each vehicle has its own femto-datacenter.

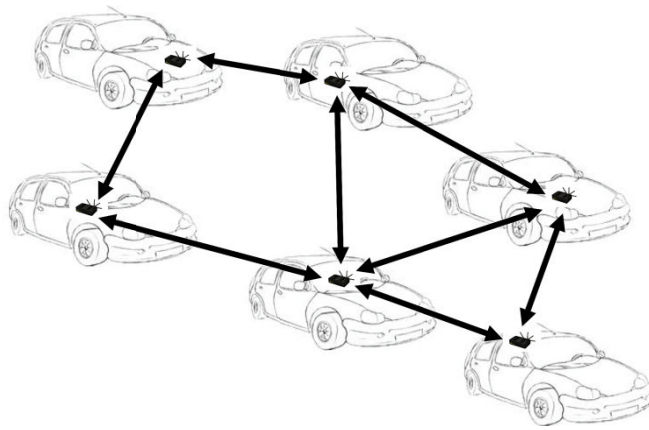


**Figure 5.5.** *Example of a mobile Cloud*

Communications between datacenters take place by way of a mesh or *ad-hoc* network. An *ad-hoc* network occurs in the case where the femto-datacenters are hosted on the same device as the communication port with the other machines possessing datacenters. In the case of a mesh network, there is a specific network which forms the link between the femto-datacenters. This network has nodes with two communication ports: one to connect with the clients and femto-datacenters, and the other with the nodes of the mesh network.

In the case of a mesh network, the vehicle has a box entirely devoted to inter-vehicle communication. The devices in the vehicle – "things", objects or any other piece of equipment – are connected to the mesh box. As we have seen, the box has two communication ports: one for connecting devices inside or outside the vehicle, and a second for communication between different boxes. The second solution – *ad-hoc* networks – can be used to form direct connections between the objects or other equipment. This solution is not advocated by operators, because the user's device needs to include networking equipment to perform communication. In addition, user applications, running on the same machine, may interfere with this equipment.

An example of a mobile Cloud is shown in Figure 5.6, where all the vehicles are traveling at approximately the same speed, which

enables the Cloudlets to simply connect with one another at acceptable speeds to form a wider Cloud. In fact, there are two mobile Clouds: one for each direction of travel of the vehicles.

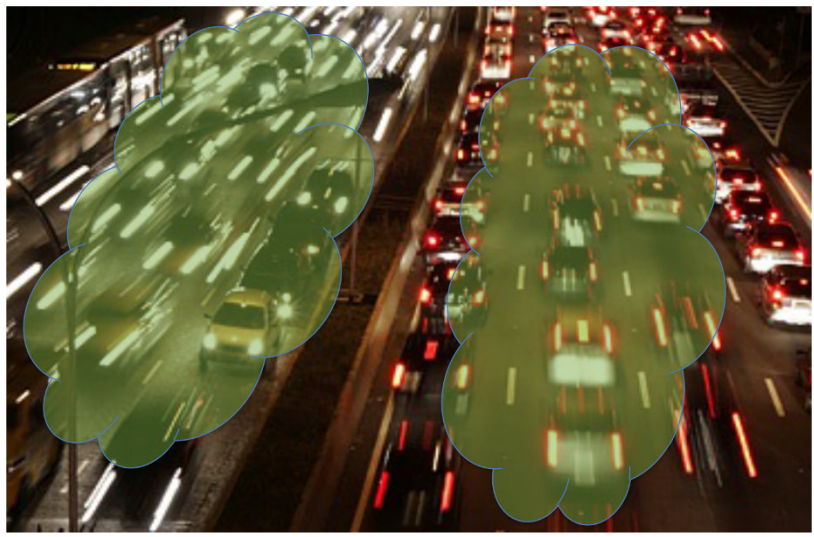


**Figure 5.6.** *Two mobile Clouds*

A large mobile Cloud can be established when a traffic jam occurs, as shown in Figure 5.7. When hundreds of vehicles are in the same place, the overall power may become quite substantial. The vehicles must be connected in a mesh network with specific protocols, such as OLSR, to allow for data transmission between the vehicles.

## 5.3. Mobility control

Mobility is difficult to manage because the decisions essentially need to be made in real time, and the motion means that it is impossible to use powerful entities such as datacenters, which are situated too far from the periphery, except perhaps if we consider new technologies such as C-RAN, where the local loop needs to be rethought, and replaced by an optical network. With few

exceptions, we must move the decisions to the endpoints, and this delocalization is done by local controllers. We again see the issue of SDN on the periphery, which we have already touched on many times. The OpenFlow protocol may play an important role in connecting the client and the controller.

**Figure 5.7.** *A large mobile Cloud*

Two types of controller may be found, low-level controllers and high-level controllers. The former handle the lower layers: frequency regulation, power regulation and adaptations of the physical and link layers – i.e. of levels 1 and 2. These controllers can also manage certain types of security applications, such as detecting hacker access points. High-level controllers are associated with SDN controllers. They are capable of handling the properties shown in Figure 5.8.

These properties contain the authentication of users by any given technique, with the harvesting of a maximal number of characteristics of the user and his/her applications. The reason for this information-harvesting is to facilitate an SDN control performed directly by the local controller or to transmit these data to a central SDN controller. The user's characteristics are often coupled with a database which is regularly updated.

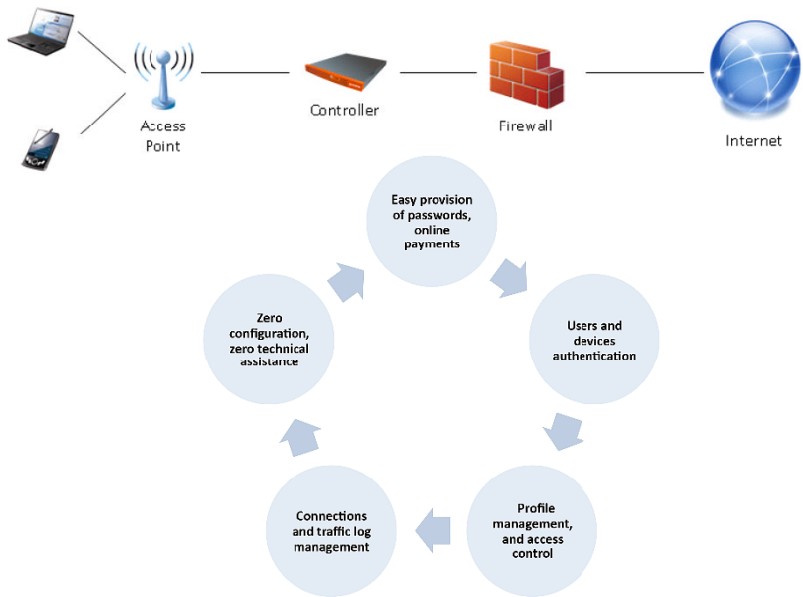



**Figure 5.8.** *Properties of mobile device controllers*

The second major property is to determine the user profile, either by harvesting information when authenticating, or by associating the user with a pre-defined profile. This profile enables us to determine the applications which can be run, and thus facilitate access control depending on the user's demands. For example, the user profile indicates whether or not s/he has access to the Internet solely via HTTP, has the right to use VoIP services, send e-mail, use a local printer, etc.

The third property of controllers pertains to the management and control of flows by using a protocol such as OpenFlow, or another, more conventional, protocol. The controller must also be capable of satisfying the obligatory demands of states, such as storing the logs of clients who connect to the controller.

The last two sets of properties pertain to the management of the connection, firstly by providing the elements necessary to perform a

"zero configuration": the controller is capable of substituting the mobile terminal to provide particular properties. For example, the controller can intercept a user's electronic messages and send them using its own SMTP, in the knowledge that the network to which the mobile is connected does not accept the messages of the user who is a guest of the company. Another example is the use of a printer from a mobile phone. The controller can load a virtual driver and enable the mobile to print locally. Finally, the controller can manage solutions to enable the user to use passwords to connect to the access point.

If we examine the authentication more specifically, the most conventional method is a window in which the client enters his/her username and password, using the IEEE 802.1x standard, for example. Another solution is access by authentication on another site, such as Google or Facebook, or on a payment server. Once the external authentication has been performed, a specific protocol such as OAuth can send the authentication back to the controller. One final solution may be to ask the user to fill out an information page in order to gain access.

It should be noted that in the last two scenarios, the client must be able to connect to the network without authentication. Thus, the access point must allow him/her to pass on a specific, provisional authorization of the controller. This solution is often an option in authentication systems. However, it is somewhat risky, because once he/she is authenticated on an external server, the client can continue to navigate without returning for a firm authentication on the local controller.

Figure 5.9 presents the two solutions for application controllers: the first is physical and the second virtual. The advantage of the virtual controller is that it uses the resources which are necessary at any given moment – i.e. a great deal of resources at peak times and practically no resources during trough periods.

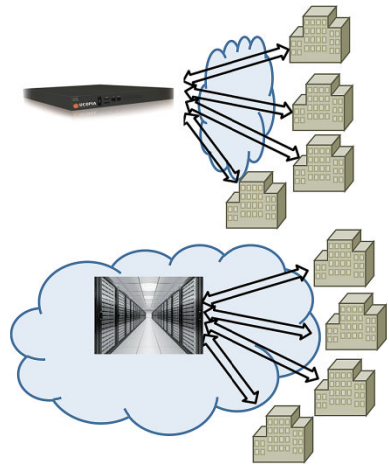


**Figure 5.9.** *The two controller solutions*

Figure 5.10 illustrates the different ways in which it is possible to obtain Internet access authorization using an access point and an access-point controller. Case 1 corresponds to client access to an external authentication site. He/she obtains provisional authorization to pass the access point and seek authentication or recognition on an external site which may be located anywhere on the Internet. Once this authentication has been performed, it is transmitted back to the controller, which sends the order to the access point to allow the client to pass.

Case 2 is that which facilitates a solution with an SDN controller. When a flow comes in, the SDN access point sends a request to the SDN controller, which begins by authenticating the client on a captive portal or using another authentication solution. Once authentication has been completed, the controller grants authorization to the access point, specifying which flow is to be allowed through. The client can then access the Internet.

Case 3 is that of a local physical controller which, after local authentication, allows the client to communicate with the Internet via the access point. The IEEE 802.1x protocol can easily be used in this situation.

Case 4 represents the solution where we need to access the central controller which, after authentication and consultation of the profile, allows the client to access the Internet.

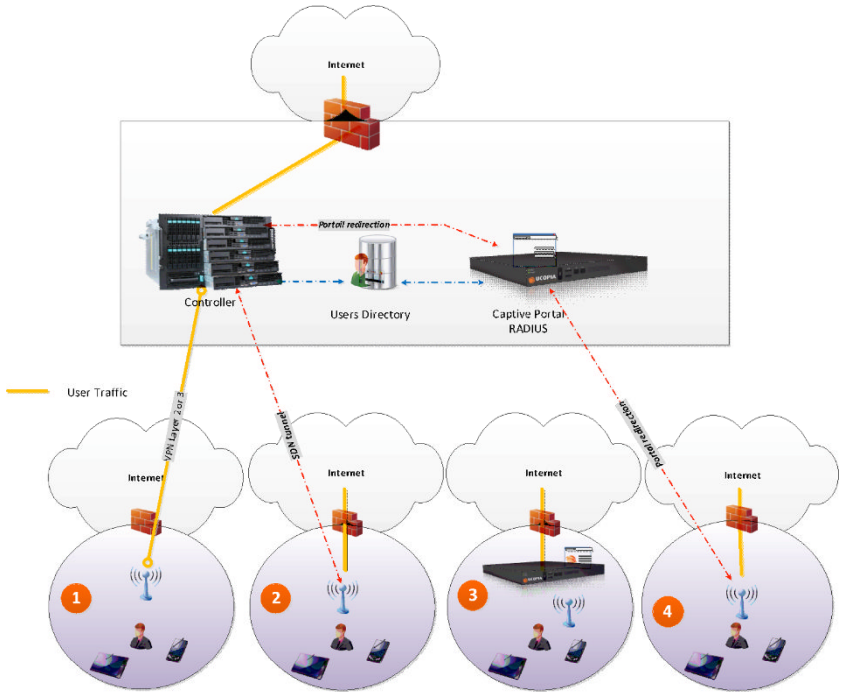



**Figure 5.10.** *Access authorization cases for a controller*

## 5.4. Mobility protocols

The convergence of landline and mobile telephones means that any application installed on a server can be used from a fixed or mobile device. The first condition to achieve this convergence is that a mobile be able to remain connected to the network whilst moving around, without the functioning of the application being interrupted. Changes of cell (known as handovers) must take place transparently and seamlessly, with the same quality of service being maintained if the application requires it.

We shall begin by discussing macromobility, which involves the management of mobility of terminals between two different IP domains, and micromobility, which relates to the change of attachment point within the same IP domain. In the first case, the main protocol is IP Mobile; in the second, various protocols are in competition with one another. We shall then go on to examine multihoming – i.e. the case where a terminal is simultaneously connected to several networks.

## 5.5. Mobility control

Two main types of mobility control solutions in IP networks have been put in place, depending on the mobile terminal's movement: support for macromobility and support for micromobility.

In the former case, the mobile changes the IP network, whereas in the latter, it stays within the same network but changes its attachment antenna. Macromobility is handled by the IP Mobile protocol. For micromobility, numerous protocols have been defined. We shall examine the main ones in this chapter.

### 5.5.1. *IP Mobile*

IP is increasingly being presented as a solution to the problems posed by mobile users. The IP Mobile protocol can be used in IPv4, but the potential lack of addresses complicates the management of communication with the mobile. IPv6 is preferable, because of the large number of available addresses, which means that temporary addresses can be assigned to the stations as they move around.

The operation of IP Mobile is as follows:

– a station has a base address, with an agent attached to it, whose role is to monitor the match between the base address and the temporary address;

– when a call comes into the mobile station, the request is sent to the database in which the base address is held;

– because of the agent, it is possible to match the base address to the provisional address and send the connection request to the mobile.

This solution is similar to that used in mobile networks.

The terminology employed in IP Mobile is as follows:

– Mobile Node: terminal or router which changes its attachment point from one sub-network to another;

– Home Agent: router of the sub-network with which the mobile node is registered;

– Foreign Agent: router of the sub-network being visited by the mobile node.

The IP Mobile environment is formed of three relatively separate functions:

– agent Discovery: when the mobile arrives in a sub-network, it searches for an agent capable of handling it;

– registration: when a mobile is outside of its home domain, it registers its new address (Care-of-Address) with its Home Agent. Depending on the technique used, registration may take place directly with the Home Agent, or be done through the Foreign Agent;

– tunneling: when a mobile is outside of its home sub-network, the packets need to be delivered to it by way of the technique of tunneling, which links the Home Agent to the Care-of-Address.

Figures 5.11 and 5.12 illustrate the arrangement of communication in IP Mobile for IPv4 and IPv6.

### 5.5.2. *Solutions for micromobility*

Various protocols have been put forward by the IETF for the management of micromobility in order to improve IP Mobile. Indeed, micromobility solutions are primarily intended to reduce the signaling messages and the latency of handover caused by the mechanisms of IP Mobile. Generally, we distinguish two categories of approaches for

micromobility: those which are based on tunnels and those which use forwarding tables.

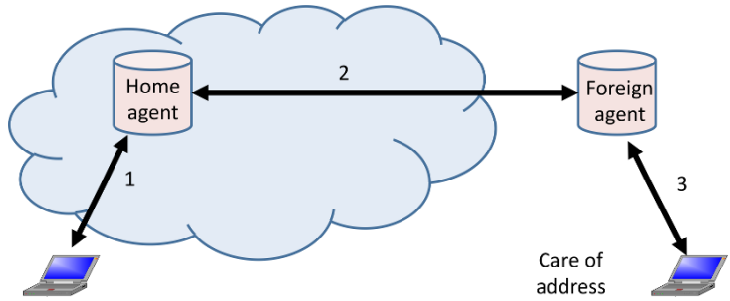



**Figure 5.11.** *IP Mobile for IPv4*

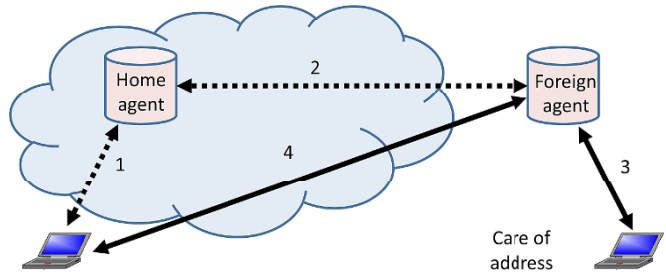



**Figure 5.12.** *IP Mobile for IPv6*

Tunnel-based approaches use local and hierarchical registration. For this purpose, they use IDMP (Intra-Domain Mobility Management Protocol), HMIPv6 (Hierarchical MIPv6) and FMIPv6 (Fast MIPv6).

HMIPv6 is a hierarchical arrangement which differentiates local mobility from global mobility. Its advantages are the improvement of handover performances, because local handovers are dealt with locally, and of the transfer rate, along with the minimization of the loss of packets which may arise during the transitions. HMIPv6 considerably reduces the signaling load for managing mobility on the Internet, because the signaling messages corresponding to local movements do not travel through the whole of the Internet, but instead remain confined to the site.

The aim of FMIPv6 is to reduce the latency of handover by remedying the shortcomings of MIPv6 in terms of time taken to detect the mobile's movement and register the new care-of-address, but also by pre-emptive and tunnel-based handovers.

Approaches based on forwarding tables consist of keeping routes specific to the host in the routers to transfer messages. Cellular IP and HAWAII (Handoff-Aware Wireless Access Internet Infrastructure) are two examples of micromobility protocols using forwarding tables.

## 5.6. Multihoming

Multihoming involves a terminal connecting to several networks simultaneously in order to optimize the communications by selecting the best network, or by allowing an application to take several paths. Thus, the packets forming the same message may take several different paths simultaneously. In the slightly longer term, a terminal will be connected to several networks, and each application will be able to choose the transport network (or networks) it uses.

The term "multihoming" refers to the multiple "homes" that a terminal may have, with each "home" assigning its own IP address to the terminal. Thus, the terminal has to manage the different IP addresses assigned to it.

Multihoming enables us to have several simultaneous connections to different access networks. A multihomed terminal is defined as a terminal which can be reached via several IP addresses.

We can distinguish three scenarios:

– *a single interface and several IP addresses*. This is the case for a terminal in a multihomed site. Each IP address of the terminal belongs to a different ISP. The goal of this configuration is to improve reliability and optimize performances by using traffic engineering;

– *several interfaces, with one IP address per interface*. The terminal has several physical interfaces, and each interface has only one IP address. Such is the case with a multi-interface mobile

terminal. The interfaces may be different access technologies and be connected to different networks. Each interface obtains an IP address assigned by its own network;

– *several interfaces, with one or more IP addresses per interface*. This is the most commonplace situation. The terminal has several physical interfaces, each of which can receive one or more IP addresses, attributed by the attached networks.

There are two main approaches to multihoming: one using the routing protocol BGP (Border Gateway Protocol), and one using the Network Address Translation mechanism. These solutions are based on network routing and address management techniques to achieve high connection ability and improved performance.

The earliest multihoming protocols were at transport level, with SCTP (Stream Control Transmission Protocol) and its extensions. At network level, SHIM6 (Level 3 Multihoming Shim Protocol for IPv6) and HIP (Host Identity Protocol) are the two protocols supporting multihoming standardized by the IETF.

Another way of looking at multihoming protocols for a mobile terminal is to develop the multipath transmission protocol on the basis of the single-path transmission protocol. Such is the case with the registration of multiple addresses in a mobility situation: addresses known as the multiple Care-of-Address (mCoA) and MTCP (Multipath TCP). The mCoA protocol is an evolution of Mobile IPv6, which allows multiple temporary addresses (CoAs) to be registered on the same mobile terminal. This solution defines a registration ID whose purpose is to distinguish the CoAs of the same terminal. MPTCP is a recent extension of TCP to take care of data transmission along several paths simultaneously. An MPTCP connection is made up of several TCP connections.

The objectives of a multihoming protocol for a multi-interface mobile terminal are as follows:

– the available paths must be simultaneously used to increase the bandwidth;

– the load-sharing algorithm which distributes the data along multiple paths must ensure that the total bandwidth is at least equal to that obtained on the best path;

– the use of several paths must enhance the reliability of the communication;

– the multihoming protocol must support mobility by dealing with the problems of horizontal and vertical handovers;

– the interfaces of different access technologies should provide users with better radio coverage.

## 5.7. Network-level multihoming

In this section, we present three protocols that take care of multihoming at network level: HIP (Host Identity Protocol), SHIM6 (Level 3 Multihoming Shim Protocol for IPv6) and mCoA (Multiple Care-of-Addresses).

Before going into detail about HIP and SHIM6, let us introduce the main principles of these two protocols. In an IP network, the IP address is used both as the terminal's identifier and locator. We have already seen this property in the case of the LISP protocol. This dual function of the IP address poses problems in a multihoming environment. Indeed, as each terminal has several IP addresses associated with its interfaces, a terminal is represented by multiple IDs. In mobility, when the terminal moves outside of the coverage range of a given technology, it switches to the interface of another technology. Communication is then interrupted because the upper layers think that the different IDs represent different terminals. The simultaneous transmission of data along several paths is impossible, because those paths are not considered to belong to the same terminal.

With HIP and SHIM6, we separate the two functions of the IP address in order to support multihoming. An intermediary layer is added between the network layer and the transport layer. This layer enables us to differentiate the node's ID and its address. The ID no longer depends on the node's attachment point. Although the node may have several addresses, it is introduced by a unique identifier in

the upper layers. When one of the node's addresses is no longer valid, the ID is associated with another address. The intermediary layer handles the link between the ID and the address. Therefore, the change of address becomes transparent to the upper layers, which associate only with the ID.

### 5.7.1. *HIP (Host Identity Protocol)*

HIP is an approach which totally separates the identifying function from the localizing function of the IP address. HIP assigns new cryptographic "Host Identities" (HIs) to the terminals.

In the HIP architecture, the HI is the terminal's ID presented to the upper layers, whereas the IP address only acts as a topological address. As is indicated by Figure 5.13, a new layer, called the HIP, is added between the network and transport layers. This HIP layer handles the matching of the terminal's HI and its active IP address. When a terminal's IP address changes because of mobility or multihoming, the HI remains static to support that mobility and multihoming.

The HI is the public key in a pair of asymmetrical keys generated by the terminal. However, the HI is not directly used in the protocol, because of its variable length. There are two formats of HI, using cryptographic hashing on the public key. One representation of HI, with 32 bits, called the LSI (Local Scope Identifier), is used for IPv4. Another representation of HI, with 128 bits, called the HIT (Host Identity Tag), is used for IPv5.

Before communicating with one another, the terminals use the HIP base exchange protocol to establish the HIP association. HIP base exchange is a cryptographic protocol which enables the terminals to authenticate one another. As shown by Figure 5.14, the base exchange comprises four messages that contain the communication context data, such as the HITs, the public key, the security parameters of IPSec, etc. I and R respectively represent the initiator and the responder of the exchange. In order to protect the association against Denial-of-Service (DoS) attacks, a puzzle is added into messages R1 and I2. As the HI

replaces the identifying function of the IP address, the HI and one of the IP addresses available for a terminal must be published in the DNS (Domain Name System).

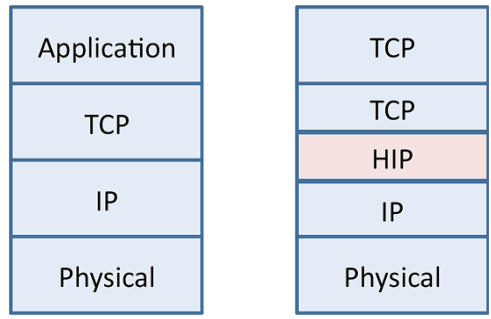



**Figure 5.13.** *HIP architecture*

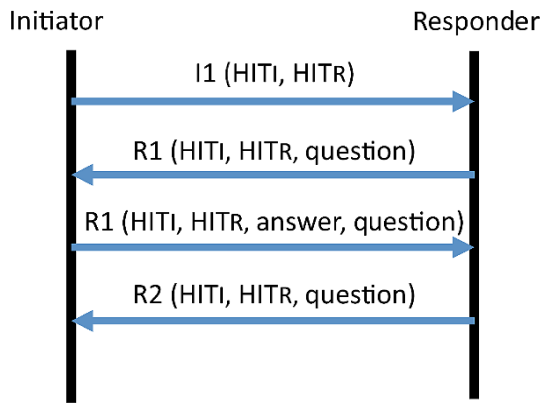



**Figure 5.14.** *Base exchange procedure for HIP*

Figure 5.15 illustrates the base exchange procedure, involving the DNS server. Before beginning the HIP base exchange, the initiator interrogates the DNS, using the responder's DNS name, to obtain its HI and its IP address. After having received the necessary data, the initiator executes the basic exchange by means of four messages: I1, R1, I2 and R2.

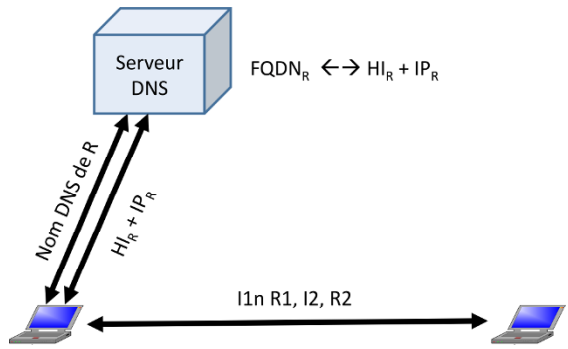



**Figure 5.15.** *HIP base exchange with DNS*

### 5.7.2. *SHIM6 (Level 3 Multihoming Shim Protocol for IPv6)*

Like network-level multihoming protocols, SHIM6 also supports the separation of the identifying and localizing roles of the IP address. SHIM6 employs an intermediary SHIM sub-layer within the IP layer.

Unlike HIP, the SHIM6 approach does not introduce a new namespace for the identifiers. The terminal considers one of its IP addresses as its identifier. Known as the ULID (Upper-Layer Identifier), it is visible to the upper layers. Once the terminal's ULID is chosen, it cannot be modified for the duration of the communication session. This mechanism ensures transparent operation of all upper-layer protocols in a multihoming environment, because these protocols always perceive a stable IPv6 address. The SHIM sub-layer handles the matching of the ULID and the terminal's active address during the transmission.

As illustrated in Figure 5.16, terminal A has two addresses – IP1A and IP2A – and so does terminal B (IP1B, IP2B). The stable source and destination addresses perceived by the upper layers are, respectively, IP1A and IP1B. Suppose that the IP1A and IP1B address are no longer valid for the two terminals. If terminal A continues to send packets to terminal B, the application of terminal A indicates IP1A and IP1B as the packet's source and destination addresses, because it is not aware of the change of addresses. When the SHIM sub-layer of terminal A receives that packet, it converts the indicated

ULIDs into the real addresses of the two terminals. Then, the packet is sent over the address IP2A of terminal A to the address IP2B of terminal B. When the SHIM sub-layer of terminal B receives the packet, it translates the addresses back into ULIDs (IP1A for the source and IP1B for the destination) before sending data up to the next layer. Due to the matching between the ULID and the address performed by the SHIM sub-layer, the changes of address are totally transparent for the upper-layer protocols.

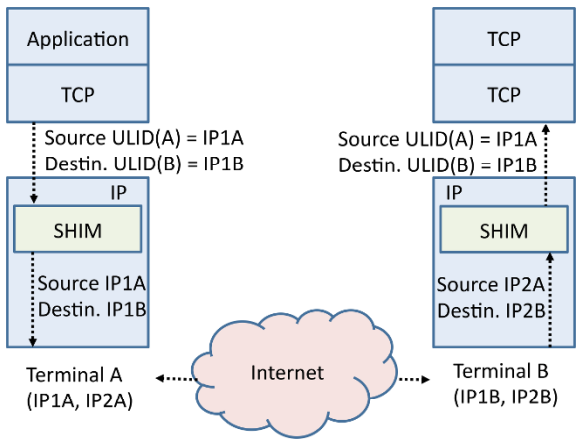



**Figure 5.16.** *Example of matching in SHIM6*

### 5.7.3. *mCoA (Multiple Care-of-Addresses) in Mobile IPv6*

MIP6 (Mobile IPv6) is the protocol used to manage mobility for IPv6. In the IP mobility architecture, each mobile node has a static HoA (Home Address) which is attributed by its home network. When the mobile node enters a host network, it obtains a new, temporary IP address known as the CoA (Care-of-Address). It informs its home agent (HA), which is in the home network. The HA stores, in its cache (Binding Cache), the match between the CoA and the HoA of the mobile node. Then, if there is any traffic destined for the HoA, it is intercepted by the HA and redirected to the mobile node's true address using an IP tunnel. Consequently, the mobility of the mobile node becomes transparent to the corresponding nodes. The principle of Mobile IPv6 is illustrated in Figure 5.17.

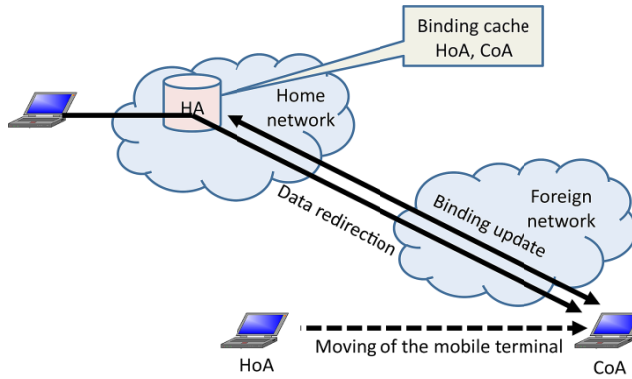



**Figure 5.17.** *Mobile IPv6*

The limitations of Mobile IPv6 stem from the fact that the client can only register one CoA with his/her HA. Indeed, when the mobile node moves to a new network, it obtains a new CoA. The mobile node must inform its HA of this change in order for the HA to update its cache. The HA then replaces the old CoA with the new one. Consequently, at any given time, there can only be one association between a CoA and HoA.

With a view to being able to support multihoming, an extension to facilitate the registration of multiple CoAs was defined: a mobile node receiving several CoAs associated with its interfaces can register the same number of matches between its CoAs and its unique HoA with its HA. With this goal in mind, a binding identifier (BID) in the cache is used. Every instance of a registration (binding), which is equivalent to a match between a CoA and an HoA, is identified by a unique BID. By using one or more Binding Update messages, the mobile node informs its HA of its CoA/HoA matches and the corresponding BIDs. In its binding cache, the HA stores all the matches bound by the mobile node, as illustrated by Figure 5.18. Multipath transmission is managed by the mobile node. For outgoing data flows, the mobile node sends the traffic in accordance with its flow distribution algorithm. For incoming data flows, the mobile node defines the

traffic preferences and informs the HA of those preferences. The HA then distributes the traffic destined for that mobile node via the different CoAs in keeping with the stated preferences.

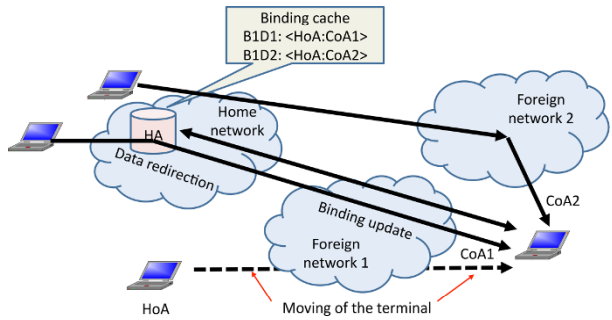



**Figure 5.18.** *Registration of several CoAs in Mobile IPv6*

## 5.8. Transport-level multihoming

In the classic transport protocols TCP and UDP, each terminal uses a single address for a communication. If the terminal's address changes, the connection is interrupted. With the evolution of access technology, it is quite common for a terminal to have several IP addresses associated with its interfaces. The solutions put forward for the transport level involve each terminal maintaining a list of IP addresses. One address can be replaced by another without the communication being broken. Transport-level multihoming protocols are extensions of TCP such as MPTCP (Multipath TCP), Multihomed TCP, TCP-MH, and SCTP (Stream Control Transmission Protocol). In this section, we shall discuss the two most widely used standardized protocols: SCTP and MPTCP.

### 5.8.1. *SCTP (Stream Control Transmission Protocol)*

Stream Control Transmission Protocol (SCTP) is a transport protocol which was defined by the IETF in 2000 and updated in 2007. Originally, this protocol was developed to transport voice signals on the IP network. As it is a transport protocol, SCTP is equivalent to other transport protocols such as TCP and UDP. SCTP offers better

reliability, thanks to congestion-monitoring mechanisms, and the detection of duplicate packets and retransmission. In addition, it supports new services which set it apart from other transport protocols: multistreaming and multihoming. Unlike TCP, which is a byte-oriented protocol, SCTP is a message-oriented protocol. This is illustrated in Figure 5.19 – the data are sent and received in the form of messages.

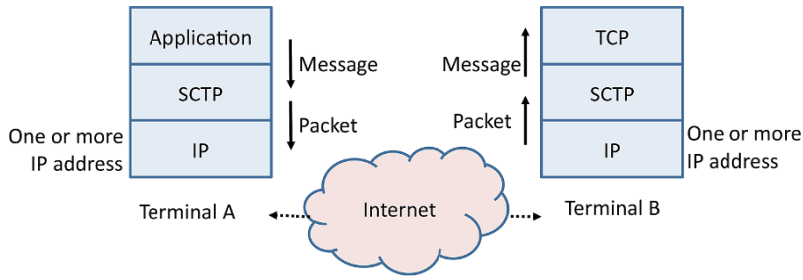



**Figure 5.19.** *SCTP architecture*

When the SCTP transport service receives a data message from an application, it divides that message into *chunks*. There are two types of chunks: those containing user data and control chunks. In order to be sent to the lower layer, the chunks are encapsulated into SCTP packets. As Figure 5.20 indicates, an SCTP packet may contain several chunks of data or control chunks. Its size must be less than the MTU (Maximum Transmission Unit).

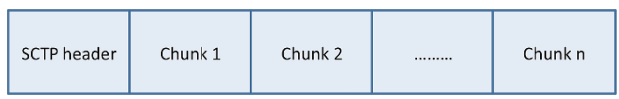



**Figure 5.20.** *Structure of SCTP packet*

The two SCTP terminals establish an SCTP connection. As mentioned above, SCTP can support multistreaming on a single connection. Unlike with TCP, several flows can be used simultaneously for communication in an SCTP connection. If congestion occurs on one of the paths, it only affects the flow being

sent along that path, and has no impact on the other flows. Each SCTP flow independently numbers its chunks, using the sequence numbers StreamID and Stream Sequence Number (SSN).

Thanks to multistreaming, transport capacity is greatly improved. One of the most important services provided by SCTP is multihoming support. In this case, each SCTP terminal has a list of IP addresses linked to its interfaces.

When the connection is established, the SCTP terminals exchange their address lists. Of the IP addresses on that list, the terminal chooses one as a primary address, with the others being secondary addresses. In accordance with this principle, each SCTP connection has one primary path and multiple secondary paths. The primary path is used for data transmission. In case the primary path breaks down, SCTP uses one of those secondary paths to continue communication.

SCTP uses the Heartbeat mechanism to check whether a path is active. The Heartbeat message is periodically sent to each address advertised by the other terminal. The consecutive absence of an acknowledgement, Heartbeat-Ack, from the addressee indicates that a path is unavailable. Figure 5.21 illustrates the Heartbeat mechanism.

Terminal A periodically sends the Heartbeat message (every THB seconds) to check the availability of the path between two IP addresses (IP1a and IP1b). After $n$ tests without acknowledgement, the path (IP1a, IP1b) is deemed to be inactive. If the primary path becomes inactive, a secondary path is used to send data until the primary path becomes active once more.

In each SCTP connection, a single path may be used for the data transmission, with the others being considered as backup paths. Multihoming stems from the possibility of using several paths simultaneously. Extensions of SCTP are needed in order for SCTP to be able to completely support multihoming.

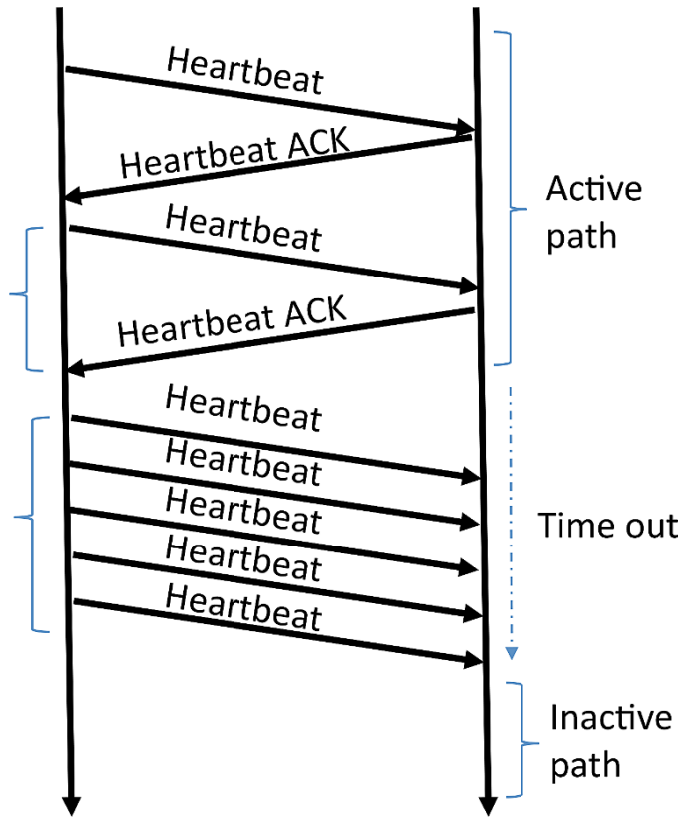



**Figure 5.21.** *Heartbeat mechanism*

To support mobility, SCTP offers an extension called Mobile SCTP, which allows dynamic address reconfiguration. When a change of IP address takes place because the terminal has moved, that terminal sends an ASCONF (Address Configuration Change Chunk) message to request the addition of the new IP address, the deletion of the old address and possibly the changing of the primary address for that connection.

LS-SCTP (Load Sharing SCTP) is another extension of SCTP which allows the aggregation of bandwidth in SCTP. The terminals can simultaneously use several available paths for the transmission, ensuring the independence of the congestion control for each path.

The extension LS-SCTP is composed of two new elements:

– a new parameter in the INIT and INIT-ACK chunks, which indicates whether the terminals support the extension;

– two new types of chunks: an LS-Data (Load-Sharing Data) chunk, used for sending data, and an LS-SACK (Load-Sharing SACK) acknowledgement chunk, which provides selective acknowledgements on each path.

Figure 5.22 illustrates the architecture of LS-SCTP. On each side of the connection, we see an overall "Association Buffer", a Path Assignment Module and a Path Monitor. LS-SCTP separates the flow control from the congestion control. The flow control is managed at the level of the connection. The terminals use the association buffer to receive data from all paths before sending them to the corresponding terminal or to the application.

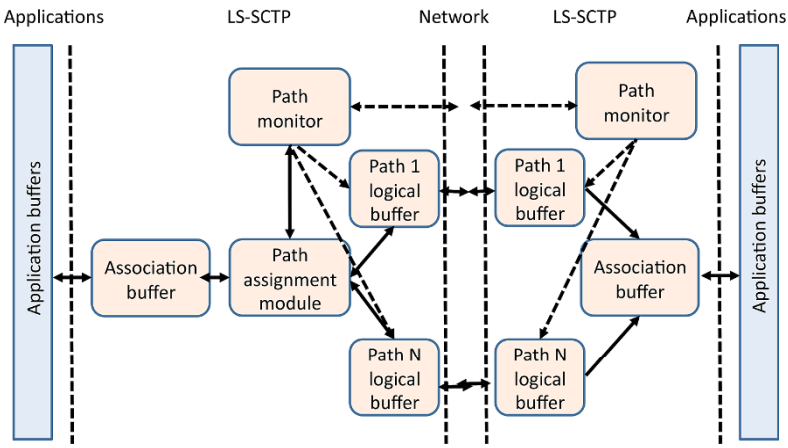



**Figure 5.22.** *Architecture of LS-SCTP*

The fragmentation and reassembling of the data take place in the connection buffer. Congestion control is performed for each path. Each path has its own congestion control parameters, such as the size of the congestion window (cwnd), the slow-start threshold (ssthresh), the round-trip time (RTT), etc.

To separate the flow control from the congestion control, LS-STCP uses two different sequence numbers. The first is the ASN (Association Sequence Number), which is the sequence number for the active connection. It is used to order all the chunks received by the buffer on the connection to the receiver. The second is the sequence number of data chunks for each path, called the PSN (Path Sequence Number). It ensures that each path is reliable and controls the path congestion. In the LS-SCTP architecture, the Path Assignment module takes care of the distribution of the data between the available paths. This distribution is based on the availability of bandwidth. This module distributes chunks depending on the cwnd/RTT ratio of each path. Once the path is selected, the chunk provides a PID, indicating the path and a PSN.

Another difference between LS-SCTP and SCTP is the retransmission mechanism. To improve the likelihood of a retransmitted data chunk reaching the receiver, the retransmission uses a different path to that used for the previous attempted transmission. The Path Monitor module is responsible for overseeing the available paths and updating the list of active paths. This module updates the active path list when it obtains information from the network relating to the failure of an existing path or the availability of a new path. When the transmitter detects that a path is unavailable, it changes the status of that path to “inactive”. Inactive paths still remain associated with the connection. The transmitter continues to monitor them by means of the HeartBeat mechanism. As soon as an inactive path becomes active once more, Path Monitor adds it to the list of active paths and uses it for transmission.

### 5.8.2. *CMT (Concurrent Multipath Transfer)*

Multihoming is incompletely supported by SCTP. The only reason for having multiple paths available is to have redundancy in place. SCTP transmits via the primary path. The secondary paths are used only if the primary path fails. An extension of SCTP, called CMT (Concurrent Multipath Transfer), has been put forward for transmission via multiple paths.

In CMT, there is no distinction between the primary and secondary paths. All paths are equivalent, and are used simultaneously in data transmission. CMT employs the same system of sequencing numbers as SCTP. CMT uses the round-robin algorithm to send the data along the available paths. The sending of data to the transmitter is monitored by the size of the congestion window (CWND) for each path. CMT sends the data along a path as soon as the congestion window for that path becomes available. When several paths can be used for the transmission, the path is selected using the round-robin model. CMT fills the congestion window of each path before moving on to the next.

Despite the benefits of using multipath transmission, CMT brings with it the phenomenon of de-sequencing, which degrades performances, because of the fact that the paths in CMT may have different characteristics in terms of bandwidth and delay.

CMT identifies three causes of de-sequencing of data at the receiver's end:

– *needless fast retransmissions*. In a multipath transmission, paths may have different bandwidths and delays. It may be that an acknowledgement from a fast path will indicate the loss of a packet while that packet is still in transit on a slower path, and only arrives at its destination later on. CMT uses the same congestion control mechanism as TCP. When the transmitter receives three duplicate acknowledgements indicating the loss of a packet, it deems that packet to be lost in the network and triggers the fast retransmission of the packet. However, this retransmission is needless, because it is the de-sequencing which is the cause of the duplicated acknowledgements;

– *inaccurate updating of the congestion window*. The mechanism by which the congestion window is updated only allows it to be increased if a new cumulative acknowledgement (CUM ACK) number is received by the transmitter. When the acknowledgements with the same CUM ACK have arrived, even if they contain new data gaps, the transmitter does not modify its congestion window. As the phenomenon of de-sequencing occurs regularly in CMT, several acknowledgements with the same acknowledgement are sent to the transmitter. When the data gaps are covered by a new

acknowledgement, the congestion window is only increased with the new data acknowledged in the most recent acknowledgement. The previously-acknowledged data in the gaps do not contribute to the growth of the congestion window. In other words, the congestion window update does not exactly repeat the volume of data transmitted;

– *increased acknowledgement traffic*. The principle of delayed acknowledgements in TCP is also used in SCTP. Instead of sending an acknowledgement for each and every packet received, the use of a group acknowledgement for several packets reduces acknowledgement traffic. SCTP uses this mechanism if the packets reach the receiver in the correct order. De-sequenced packets must be acknowledged immediately. However, as it is quite common for de-sequencing to occur in CMT, if the receiver cannot delay the sending of the acknowledgements, then acknowledgement traffic is greatly increased, which may impact the network's performance. CMT includes the following solutions for these problems:

- in order to prevent needless retransmissions, CMT offers the algorithm SFR (Split Fast Retransmit), which enables the transmitter to correctly interpret duplicate acknowledgements. SFR defines a virtual buffer for each destination within the transmitter's retransmission buffer. With the additional information of each destination, such as the highest acknowledged TSN for each destination, the transmitter can distinguish between de-sequencing and actual data loss, with a view to correctly triggering fast retransmission. A chunk with the TSN *T* for destination M is deemed lost if and only if *T* is lower than the highest acknowledged TSN for destination M;

- the algorithm CUC (Cwnd Update for CMT) is proposed to correctly update the congestion windows for the paths. At the level of the transmitter, each destination has a variable known as the PSEUDO-CUMACK, which represents the smallest anticipated TSN. Upon receipt of a SACK acknowledgement, the transmitter checks whether there is a change of PSEUDO-CUMACK for each destination. An increase of a PSEUDO-CUMACK triggers the updating of the congestion winder of the corresponding destination, even if the CUM ACK does not advance. Thus, the congestion window for each destination increases in parallel to the acknowledged data, without having to wait for the new CUM ACK;

- CMT offers the algorithm DAC (Delayed ACK for CMT) to reduce acknowledgement traffic. DAC enables the CMT to delay the acknowledgement, even if the packets arrive out of sequence. As the sending of acknowledgements is often reported, the transmitter must closely analyze each acknowledgement received to detect any loss of data as quickly as possible. For this reason, in each acknowledgement, the receiver informs the transmitter of the number of data packets it has received since the sending of the least acknowledgement. This proposition requires modifications to be made to both transmitter and receiver.

### 5.8.3. *MPTCP (Multipath TCP)*

TCP is the most widely used transport protocol by Internet-based applications, but this protocol does not support multihoming. In spite of the existence of different paths, each TCP connection is limited to serving only one path for transmission. The simultaneous use of several paths for the same TCP session could improve data rate and offer better performances by strengthening the reliability of the connection. Multipath TCP, defined by the IETF, is a modified version of TCP which supports multihoming and allows the simultaneous transmission of data along several paths. As MPTCP is compatible with TCP, there is no need to modify existing applications to use its services. MPTCP is designed to gather together several independent TCP flows in a single MPTCP connection, ensuring transparency for the upper layers.

## 5.9. Conclusion

Mobile Cloud Networking is a paradigm which is growing enormously in importance, with the development of mobile terminals which are often not very powerful in comparison to the demand of certain applications. Mobile Cloud is another paradigm where clients come together to form a Cloud. As yet, this solution is not widely used, but it is likely to increase with the installation of more powerful boxes in vehicles.

Finally, we have examined network application controllers, which, along with SDN access points, are becoming increasingly common, and this solution represents the likely future of the discipline.

# 6

# Wi-Fi and 5G

5G has not yet been standardized, and this standardization is not expected to come into force before 2020. However, it is possible, even now, to obtain a fairly clear picture of what 5G will offer, by looking at the numerous R&D projects in this field. Figure 6.1 represents the different steps to achieve 5G. Starting from the origin of mobile networks, there was analog circuit-switched 1G, then digital circuit-switched 2G, followed by digital packet-switched 3G, with the exception of operator voice signals. With 4G, we see the move to 100% IP technology, even for voice signals, and 4G also handles multimedia applications. Finally, 5G offers increased data rates, but above all, allows billions of things to be connected.

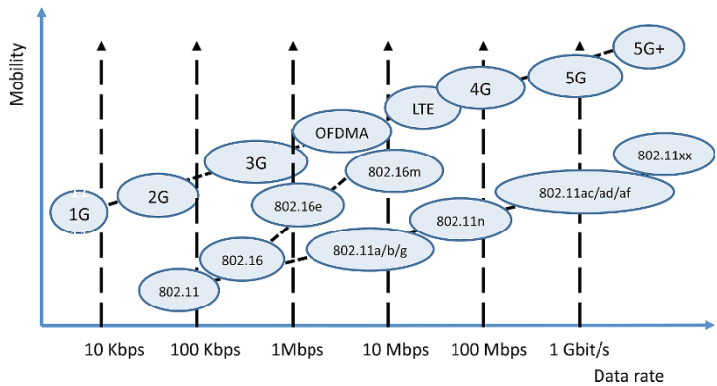



**Figure 6.1.** *The different wireless solutions*

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

In Figure 6.1, we also see two other directions taken by mobile networks. The first is WiMAX, normalized by the IEEE 802.16 working group. However, this initiative failed – not so much from the technological standpoint, but definitely so from the political perspective, given the power of mobile networks operators who have chosen the 3GPP as a standardizing body, and the trend visible across 1 to 5G. The second direction is Wi-Fi, which exhibits very promising behavior, and sells hundreds of millions of components every single day. The Wi-Fi family is currently expanding to include increasingly high data rates and increasingly longer ranges. We are going to examine these new standards in detail in this chapter.

## 6.1. 3GPP and IEEE

The 3GPP and the IEEE are the two main standardization bodies. The different standards are marked in Figure 6.2. It should be noted that LTE (Long Term Evolution) is the last version of 3G, and does not belong to 4G. The 4G series begins with LTE-A (LTE-Advanced), which is the first 100% IP standard of its kind. In LTE, telephone signals still use GSM or CDMA – i.e. 2G in digital circuit-switched mode. However, it is possible to use telephony in VoIP mode over the data part of the LTE interface. In this case, the voice signals are referred to as VoLTE (Voice over LTE).

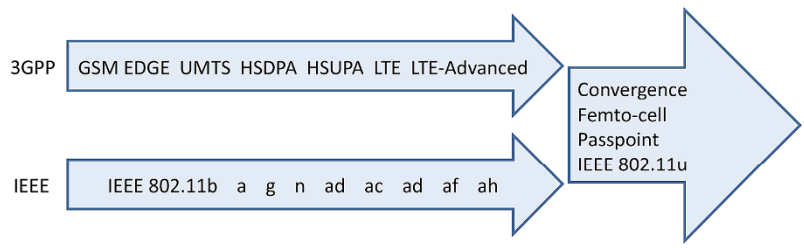



**Figure 6.2.** *The two major wireless solutions and their convergence*

Figure 6.2 shows the convergence of the two techniques developed by the 3GPP and the IEEE. In Chapter 1, we saw that only by combining these two technologies will we be able to supply the capacity that users today demand. Wi-Fi is becoming the offloading

technique for 4G/5G – i.e. it is used to relieve the load on the 4G/5G networks, with Wi-Fi antennas being used alongside the mobile network antennas. Predictions are that offloading will come to represent three quarters of the data rates of mobile terminal connections. In addition, Wi-Fi represents high data rates across hotspots, but using the Internet rather than the mobile access networks. The version of Wi-Fi for the world of telecommunications can take a variety of forms. For the moment, Wi-Fi Passpoint is the version which seems to be the most popular. Passpoint is championed by the Wi-Fi Alliance, which has already shown its strength, promoting the use of Wi-Fi through proving the conformity of products. However, other solutions are also being studied by the standardization organizations, such as replacing Wi-Fi with LTE antennas and equipping all terminals and servers with LTE ports, referred to as Home eNodeB (HeNB). We shall see that this solution fits into the context of software networks, with the dawn of software LTE.

We shall begin by examining the new generations of Wi-Fi, which have the necessary capacity to deal with the exponentially-increasing demand of users.

## 6.2. New-generation Wi-Fi

The new-generation of Wi-Fi began with the IEEE 802.11ac standard, which was released in late 2013 and achieves data rates higher than 1 Gbps. This minimum value of 1 Gbps marks the new generation of Wi-Fi. The IEEE 802.11ac standard, which we shall go on to examine in detail, is supplemented by IEEE 802.11ad, which uses a new frequency band: 60 GHz. This solution is primarily designed for the inside of buildings, and for very directional use outside. Indeed, at these frequencies, the signal is very vulnerable to interference from rain, dust and, of course, obstacles. Thus, to use it outside, we need a direct path to the access point, with good weather conditions if possible, or at least a short enough distance for the data rate to remain correct, even in the presence of rain or fog.

The new generation continued with IEEE 802.11af, which uses cognitive radio – i.e. the possibility of using frequencies for which

there is a proprietor (also known as a primary) who is not using them at a given time. For this to work, we must either drastically limit the range so as to remain in an environment without conflict with the primary, or have a server which records the use of frequencies and enables the cognitive access points to discover the usage sites and use the frequencies with no danger of interference. The IEEE 802.11af standard uses television bands, and is known as TVWS (TV White Space) or sometimes Super Wi-Fi.

There are still numerous developments under way in the field of new-generation Wi-Fi, such as IEEE 802.11ah, which pertains to long-range Wi-Fi of around a kilometer, but with much lower data rates. Its uses are mainly in the domain of the connection of "things", such as intelligent electricity meters or healthcare equipment. Numerous other standards are in the reflection phase; one of these is IEEE 802.11ax, which is an improvement of IEEE 802.11ac, with four times the capacity.

## 6.3. IEEE 802.11ac

IEEE 802.11ac is a version of Wi-Fi designed to surpass 1 Gbps, using multiple antennas, which can run simultaneously but in separate sectors. Two mutually complementary solutions have been put in place to obtain the target data rates. The first, which is fairly simple, consists simply of increasing the capacity of the transmission channel using the 5 GHz band, which is much freer than the 2.4 GHz band. The range of this higher-frequency band is a little less. In addition, obstacles hamper the signals somewhat. However, the available bandwidth is 200 MHz, which is much greater than the 83.5 MHz assigned to the 2.4 GHz band, which enables us to have Wi-Fi channels whose band is 80 MHz, and optionally, 160 MHz.

The second solution pertains to the directionality of the signals emitted, using a set of specific antennas. This technique involves allowing the transmission of several communications on the same frequency, but in different directions. There is spatial multiplexing, hence the name of this technique: SDMA (Space Division Multiple Access). Figure 6.3 illustrates this technology.

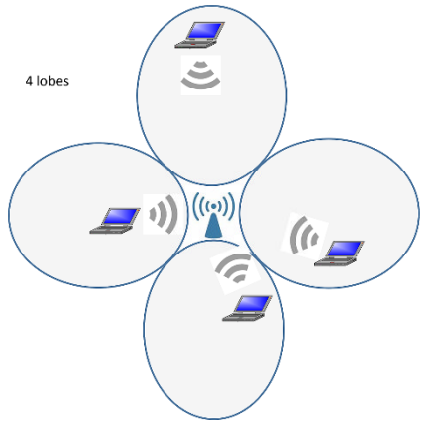



**Figure 6.3.** *SDMA*

SDMA technology uses transmission techniques allowing us to direct the signals – better known as "beamforming" – whereby the signals are orientated in a specific direction, thus giving rise to directional antennas. The set of antennas, also known as virtual antennas, enables us to generate several directional antennas simultaneously, which facilitates communication with multiple clients at the same time. The technique used in IEEE 802.11ac is also called PU2RC (Per-User Unitary Rate Control). 802.11ac-conforming antennas are also able to deliver MU-MIMO (Multi-User MIMO) – that is, to connect several pairs of clients to one another, using several antennas simultaneously on the same frequency, whilst minimizing interference. In other words, each directional antenna can be replaced by several antennas with the same directionality, using the same frequency to carry out communications in MIMO mode.

With this technology, we are able to multiply the data rate by the number of virtual antennas. In the example shown in Figure 6.3, the two physical antennas each give rise to three virtual antennas, which is a total of six virtual antennas. These six antennas mean that six simultaneous communications can be established. With these six antennas, we can also establish three 2×2 MIMO communications (two virtual antennas communicating with two antennas on a remote machine). We can just as well have three simultaneous

communications: one 3×3 MIMO communication, one 2×2 MIMO communication, and a one-way communication.

The peak data rate reaches 250 Mbps per virtual antenna. With four physical antennas, each bearing three virtual antennas, we obtain twelve virtual antennas. Thus, the total data rate is 3 Gbps. Also, if the bands used are wider – say, 80 MHz instead of 20 MHz – then theoretically we can achieve over 10 Gbps. In actual fact, as for all Wi-Fi access points, the peak data rate is only achieved in exceptional cases, where there is no external interference and the clients are located very near to the antenna. In reality, the nominal data rates are often divided by at least a factor of 2, but often much more, to obtain the real data rates.

The group IEEE 802.11ac, with 433 Mbps per antenna over an 80 MHz band and a MIMO with two antennas per client, delivers 866 Mbps per connection with a terminal. With four physical antennas, we achieve over 3 Gbps. With a bandwidth of 160 MHz, the data rate is doubled. The standard specifies that there may be up to eight spatial sectors, which again doubles the data rate. Evidently, these reported figures are peak data rates, which are only very rarely achieved in reality. As is the case with all Wi-Fi solutions, fallback rates become necessary when the conditions are not ideal. Compatibility with existing standards also requires the adaptation of data rates to conform to standards which are not compatible with IEEE 802.11ac.

The aim with the new IEEE 802.11ax standard is to further increase the data rate by using more numerous and more directional antennas.

## 6.4. IEEE 802.11ad

The IEEE 802.ad standard is completely new, with frequencies in the range of 60 GHz. This standard is championed by a group of manufacturers: the WGA (Wireless Gigabit Alliance). The name of the product, which is likely to become standard in personal area networks, is WiGig. However, the basic product could be tri-band WiGig, operating simultaneously on the 2.4 GHz, 5 GHz and 60 GHz

bands, and therefore able to adapt to the environment. The peak data rate is 7 Gbps. The range in the 60 GHz band is less than 10 meters, so this network is said to belong to the category of PANs (Personal Area Networks).

The channels used in the base mode correspond to 57.24 GHz, 59.4 GHz, 61.56 GHz and 63.72 GHz. The bandwidth is 2.16 GHz.

To compensate for the significant attenuation of the signal, IEEE 802.11ad requires directional antennas which can focus the radio beam within a 6° angle. With this very narrow beam, the IEEE 802.11ad standard is used outdoors between directional antennas with a direct view, or indoors but counting on ricochets to reach the addressee. Indeed, with a 6-degree beam, the connection between two machines is often complex. Fortunately, there is an option which enables us to direct the beam so that the two machines wishing to connect can communicate with one another.

Two applications of IEEE 802.11ad are mentioned in the standard. The first corresponds to wireless connections for peripheral computers, with the aim of doing away with the numerous cables which trail everywhere. This application also facilitates simple sharing of peripheral devices. The second application relates to consumer electronics, associated with wireless technology, such as stereo equipment, HD televisions, and online gaming systems.

## 6.5. IEEE 802.11af

The IEEE 802.11af technique pertains to a totally different method to increase the data rate: the use of cognitive radio. It consists of reusing frequency bands which are not being used at a time $t$ in the television bands.

More generally, recent measurements show that, in spite of their scarcity and their high price, frequencies between 0 and 20 GHz are under-used – often less than 10%.

The only bands which are heavily used are those employed by telecom operators which, owing to the TDMA and CDMA techniques, exhibit excellent degrees of utilization. The bands reserved for TV channels, which are below 1 GHz, are particularly attractive to operators. These TV channels could be used by access points to achieve ranges of hundreds of meters. Such ranges demonstrate that Wi-Fi technology is capable of serving the needs of WiMAX and 4G clients.

The WiFi standard IEEE 802.11af – which must not be confused with IEEE 802.3af, or PoE (Power over Ethernet) – is called TVWS (TV White Space). The expression "white space" refers specifically to those frequencies which are not used by wireless television channels. The operational bandwidth for cognitive radio covers dozens of digital television channels. The data rates achieved by this method may be very high indeed. The name of the product is "White-Fi", also sometimes called "Super Wi-Fi".

One important question relates to the way in which cognitive radio is used. It is for each state to regulate this usage. The main solutions pertain to the way in which unoccupied frequencies are used. A number of solutions suggest themselves, depending on the coverage of the cognitive access point. If the band is not occupied at the access point's location, it can be assumed that it is not occupied within a radius of a few meters of that point. This category includes Wi-Fi access points inside buildings with limited power, only covering the inside of the building. A second solution is to refer to a bandwidth management server, which knows where the bands are available. Thus, a common discipline needs to be used between the primary and secondary users. With this in mind, though, will TV operators manage to reach an agreement with Wi-Fi 802.11af users? Standardization work is being carried out in different working groups, such as IEEE P1900, with a view to proposing solutions acceptable for both parties. A third, more remote, solution could stem from the ability for each device to measure the interference and indicate to the secondary users whether or not they can exploit the frequency.

At the physical level, the method used in IEEE 802.11af is FSS (Fixed Subcarrier Spacing) with OFDM. The channels used in OFDM

may or may not be contiguous – that is, they may belong to television channels that are directly beside one another or that are separated by active television channels.

A channel contains 64 bearers. Four channels may be selected, each operating at 6, 7 or 8 MHz. In the latter two cases, as the data rate per channel reaches 26.7 Mbps and given that, as with IEEE 802.11ac, we can have at least four directions in SDMA, the maximum base rate is 568.9 Mbps, with 8 MHz channels.

The bearer access technique is the same as with normal Wi-Fi. Service classes using an EDCA (Enhanced Distributed Channel Access) technique from IEEE 802.11e are available, and correspond to the four classes Background, Best-Effort, Video and Voice. A fifth, even higher, priority is added for spectrum-monitoring, to determine which channels could potentially be used.

## 6.6. IEEE 802.11ah

The IEEE is working on a new standard, apt for the connection of "things" – i.e. sensors or other small devices with low consumption and often minimal cost. This solution works in the non-licensed bands available below 1 GHz. The technique should enable us to connect a large number of things with low consumption. The sharing of the transmissions takes place in classic Wi-Fi mode, but with a much lower data rate and a much greater range, of up to 1 km. The things are grouped into clearly-defined sets so as to avoid too high a number of simultaneous accesses and so as not to lose too much in contention in CSMA/CD mode. In addition, IEEE 802.11h involves the use of classic techniques in the world of Ethernet to put stations on standby and reawaken them.

IEEE 802.11h introduces the function of relay, whereby certain things can connect through the relay, which decreases competition by limiting the number of objects attempting to transmit simultaneously. Relay increases the geographic scope of the things that can be connected and, at the same time, saves energy by having tasks

managed locally by the relay. In the standard, the aim is not to create a mesh network, as the number of hops is limited to two.

Energy savings are also of crucial importance in the current context, where consumption is increasing exponentially, and the IEEE 802.11ah standard does take steps in this direction. For this purpose, the connected machines are divided into two classes: TIM (Traffic Indication Map) terminals and non-TIM terminals. TIM terminals periodically receive information about the traffic on the access point. Non-TIM terminals use the new mechanism TWT (Target Wake Time) to reduce the overhead of the signaling procedure.

TWT is a function which allows an access point to define a time, or a set of times, for certain terminals to have access to the communication bearer. The terminal and the access point exchange information including the allotted duration of the activity, to enable the access point to control any competition and clashes between competing terminals. The access point can reserve time periods for a terminal to transmit thanks to various protective mechanisms. The use of TWT is negotiated in advance between an access point and a terminal. The TWT mechanism is also used to reduce energy consumption, because stations using it can enter a semi-sleep state until their TWT comes around.

Another important mechanism in IEEE 802.11ah is the RAW (Restricted Access Window), which limits the number of terminals which have the right to use the access point. This limitation is achieved by putting the terminals into groups. Channel access is restricted solely to those terminals belonging to a given group, over the course of a set period of time. This mechanism helps reduce competition and avoid simultaneous transmissions from too high a number of machines which are unaware of one another's presence.

The IEEE also defines a TXOP (Transmission Opportunity), wherein an access point and a client can carry out a set of exchanges during a reserved period. The purpose of this mode of operation is to reduce the number of competing channels, improve the efficacy of the channels by minimizing the number of exchanges and help extend the

terminals' battery life by minimizing their periods of activity; the rest of the time, the terminal is on standby.

The division of the coverage area of a Basic Service Set (BSS) into sectors, determining subsets of terminal machines, is called "sectorization". It is done by a set of antennas or a set of directional antenna lobes to cover different sectors of the BSS. The goal of sectorization is to decrease contention and interference by reducing the number of terminals that can transmit simultaneously.

## 6.7. Small cells

"Small cells" is the term used to denote the new generation of cells of very small dimensions. There are numerous factors which weigh in favor of this solution, starting with reuse of frequencies, adaptation of the cells to the size of a dwelling, lowering of energy consumption, etc. Small cells are found in a variety of forms in network technology: femtocells, which correspond to a home; metrocells, which provide coverage in the street; hotspots, which are set up in public spaces; or picocells, for companies.

The main rollout should take place between 2016 and 2020. The first small cells appeared in 2010, and have become increasingly common from 2012 onwards. In certain countries, such as the United States and South Korea, there are at least three telecom operators already commercializing femtocells.

In this section, we begin by examining femtocells, which are devoted to connecting users in their homes and in the vicinity. We also present hotspots, followed by picocells, and finish up with a discussion of metrocells and microcells. We then go on to describe the Passpoint technology, which is able to bring together all of these smaller cells. In the next section, we shall look at backhaul networks, the aim of which is to connect small cells to the core network. Finally, we shall close this chapter by presenting some relevant aspects of Software Radio, Cognitive Radio and the new generation of cells.

Figure 6.4 represents a small-cell network and the backhaul network.

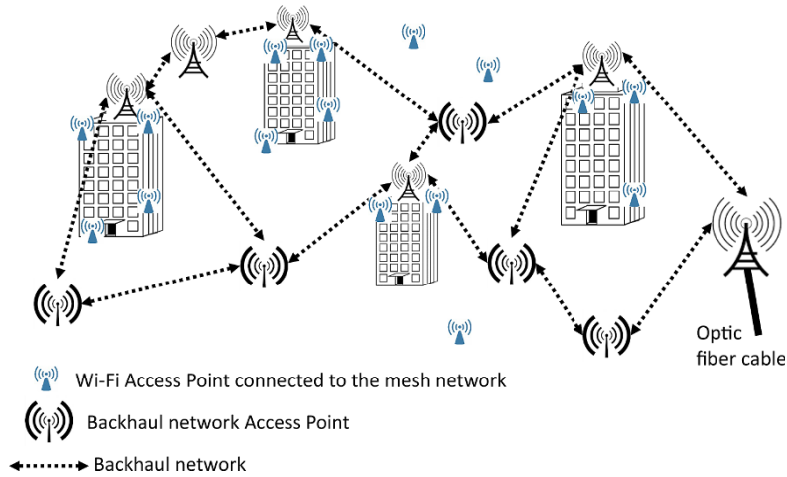



**Figure 6.4.** *Small cells and backhaul networks*

## 6.8. Femtocells

The solution advocated since release 9 of UMTS by the 3GPP to handle the enormous data rates of 4G is the use of femtocells. The fundamental ideas are fairly similar to the relay technique: so as not to have to construct a new, very dense network to connect the antennas, we use the fiber-optic infrastructure that has been in place since 2009 for very high data rate. In actuality, this optical network is not only designed to provide users with high data rates, but indeed to deliver the extremely high data rates proposed by 4G, as users cannot really tell the difference between a top-of-the-range ADSL modem and an optical modem, owing to the controls in place in the world of IP, and in particular, slow-start.

A femtocell is an antenna mounted on the user's Home Gateway. It serves all potential clients who are within the antenna's coverage area, which may be around ten meters, or indeed much more, depending on the obstacles and interference present. Thus, the femtocell is used both by the owner of the Home Gateway and by the visitors to the cell.

The obvious advantages offered by this technology are the multiplication of the cells and the densification of the network. Another very clear advantage is the drop in power of the devices which this technology facilitates. A maximum power of 100 mW seems to be the standard that is coming to pass; this power corresponds to the power required for an omnidirectional Wi-Fi transmission.

Figure 6.5 illustrates the way in which a femtocell works.

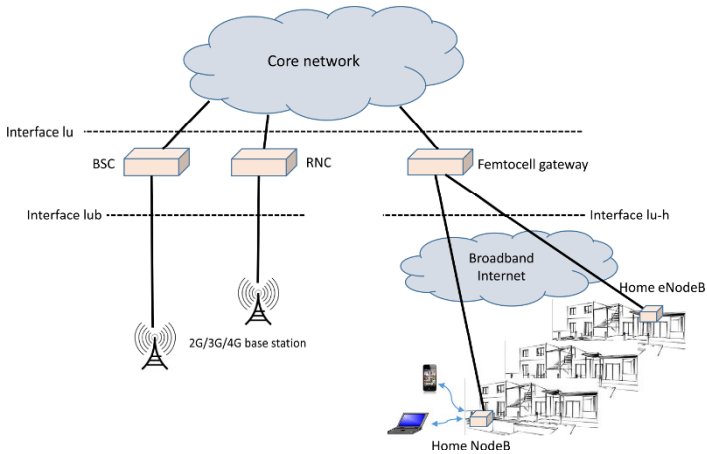



**Figure 6.5.** *Operation of a femtocell*

Conventional antennas are linked to the core network by BSC or RNC switches, corresponding respectively to 2G and 3G. The femtocell antenna is connected to a Femto-Gateway, which provides access to the core network. The connection between the Home Gateway and the Femto Gateway uses fiber-optic technology, and also often an ADSL connection. The connected terminals employ 3G, 3G+ (LTE) or 4G. The data rates are, as yet, fairly limited by radio constraints or the line to the Femto-Gateway, but by 2020, the data rates available to 4G/5G mobiles will have reached 100 Mbps.

Questions arise as to the usefulness of a Wi-Fi antenna, the possibility of a war between Wi-Fi and 4G or the supremacy of Wi-Fi

at the center of the femtocell. In fact, there is little chance of 4G antennas completely replacing Wi-Fi antennas. For that to happen, all computerized devices, printers, PCs, multimedia equipment, etc., would need to have a 4G port. Thus, there is a high probability that the two systems will coexist, with each of them handling those devices which are most appropriate.

The numerous femtocell products which are coming on the market offer diverse, varied options, as illustrated in Figure 6.6. The Home Gateway is usually referred to as an "HNB" (Home NodeB) or HeNB (home eNodelB), provided it is available with a 3G or 4G antenna. Today, these boxes essentially serve to connect 3G/4G mobiles, enabling the operator to offer the client the possibility of using his/her 3G/4G mobile in a location where there is no signal.

The HNB/HeNB very often has two radio interfaces: Wi-Fi and 3G/4G. The telecom equipment is plugged into the 3G/4G antenna, and the computer equipment into the Wi-Fi. The fiber-optic or ADSL modem simultaneously transports the flows from both environments. The second solution shown in Figure 6.6 corresponds to the use of Wi-Fi to connect telecom equipment and computers. In this case, the 3G/4G frame is encapsulated in a Wi-Fi frame and decapsulated in the Home Gateway. The 3G/4G frame is then directed to the RNC for forwarding to the core network. This solution is called UMA (Unlicensed Mobile Access) in general, GAN (Generic Access Network) for 3G, and EGAN (Enhanced GAN) for 4G.

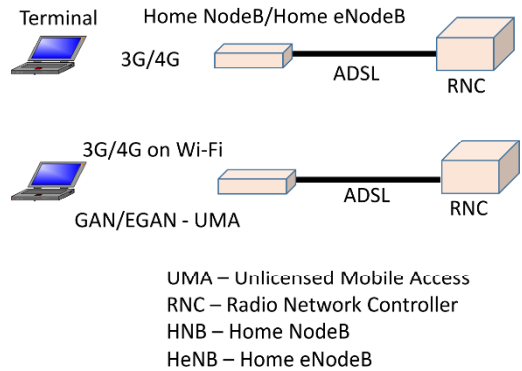



**Figure 6.6.** *Access to the HNB*

## 6.9. Hotspots

Hotspots are Wi-Fi access points whose purpose is to facilitate a high-data-rate Internet connection from a smartphone, tablet or PC. Situated in places with a great many people passing through, they are open to all clients who wish to connect, or sometimes only to those who have a subscription. If the access point is open, any and all clients can connect. However, in numerous countries, the operator of the access point must be able to monitor the clients, either in order to charge them for access or to preserve their information to locate the clients later on if the connection is used in the commission of a crime. Hotspots are located in rail stations, airports, shopping centers, at marinas and beaches, boutiques, etc.

Hotspots essentially use Wi-Fi technology and depend increasingly on telecom operators to handle their 3G/4G traffic to relieve the relay antennas which, today, are very often overloaded. This is the technique which we looked at earlier: offloading.

The difficulty for a hotspot is to manage to offer QoS to the clients' applicational flows. More specifically, the capacity of the access point depends on how far away the client is and on the degree of interference: the further away a client is, the more the signal is attenuated and the greater is the risk of interference, and therefore the access point's overall data rate degrades. In order to deal with these problems, Wi-Fi technology relies on oversizing. Still rarely available when many clients are connected to the same access point, oversizing should soon be introduced generally, thanks to new technologies which have reached maturity since 2014: specifically, IEEE 802.11ac, ad and af, which should help to ease communication.

In today's world, operators use Wi-Fi for the purposes of offloading, but as the solution is not standardized, each operator has developed their own technique (although they are all fairly similar to one another). Since 2014, the technology championed by the Wi-Fi Alliance – Passpoint – has served as a common standard for all operators. This technology is described in detail a little later on.

Metrocells are small cells which are installed in the streets to handle the 3G/4G traffic of operators. They are the equivalent of femtocells, but instead of being situated in the private domain of the home, they are in the public domain. They need to cover a surface area slightly larger than femtocells. Indeed, as there are fewer obstacles in the public domain than in the home, the clients should be able to connect, provided they have a direct view of the access point. The size of the cell is a few tens of meters. A network of metrocells is illustrated in Figure 6.7, with the metrocells serving for offloading – i.e. taking the strain off the main 3G/4G antenna.

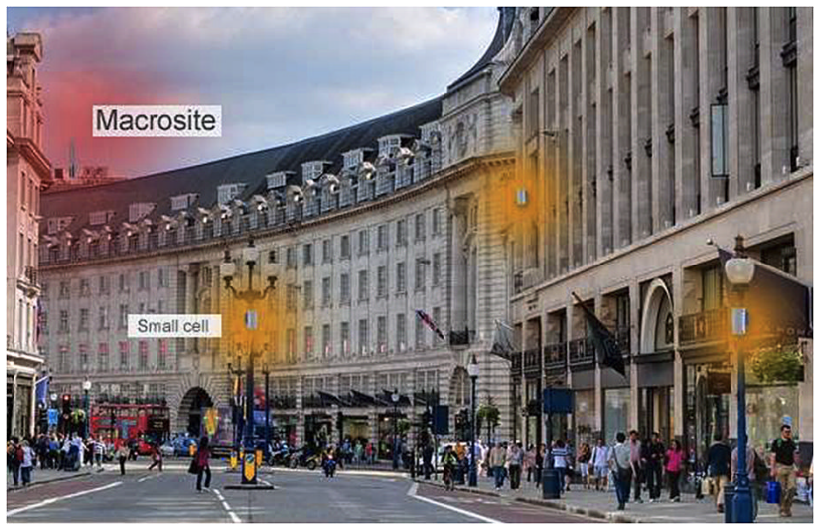



**Figure 6.7.** *A network of metrocells*

The purpose of these metrocells is to always allow offloading – i.e. to handle 3G/4G communications for the operators in order to alleviate the workload of the large antennas. The metrocells are either interconnected to form a mesh network, or connected by optical fibers, but the cost of implementing the latter solution may prove prohibitive. They may also be linked by PLC (power-line communication) as in Figure 6.7, where the access points are mounted on lampposts.

## 6.10. Microcells

Microcells are designed for use by companies. They can serve the company's internal network provide access to its intranet whilst also allowing telecom equipment (smartphones, tablets, etc.) to connect directly to the operators' networks. In other words, they must deliver two forms of access simultaneously, which may be achieved by two distinct SSIDs on the same access point, or else by using virtualization. By virtualization, we are able to put in place two virtual access points on the same physical access point. In both cases, the two networks – the company's and the network that connects to the operator network – need to be perfectly separated from one another; this is known as isolation of the two networks.

Microcells also use Wi-Fi techniques, and rely on the new standards to deliver the quality of service necessary to handle telephony and video with a good level of quality.

Engineering plays a very important role for the installation of microcells, because a company needs to cover all of its offices, workshops and boardrooms, with all of the problems caused by walls, floors and other obstacles. It is essential, in this context, to minimize interference between the access points, whilst ensuring overall coverage with as high a data rate as possible.

## 6.11. Wi-Fi Passpoint

Passpoint is an architecture proposed by the Wi-Fi Alliance. The Wi-Fi Alliance was set up when Wi-Fi was first proposed with the aim of promoting this technology. It has had a significant part to play in the success of Wi-Fi and the introduction of various standards, such as WPA2, which offers high security in Wi-Fi networks, or the introduction of priorities with IEEE 802.11e.

Passpoint is a solution which helps to relieve the workload of 3G/4G antennas by offloading, giving users the option of connecting in a way that is totally transparent to the 3G/4G networks, using Wi-Fi networks. In other words, the client will not know how s/he is

connected to the operator network: by a BTS, a NodeB or an eNodeB or by a Wi-Fi access point (hotspot, femtocell, microcell, metrocell, etc.).

In home or business environments, connection to a Wi-Fi network is generally made automatically once the user has entered the authentication data upon first connecting to the network. When s/he is connected to the access point and authorized to enter into the network, s/he is subject to the rules established by the IT manager or the owner of the access point. Once recognized, his/her terminal automatically joins the access points to which it connects regularly, without intervention on his/her part.

The connection to most hotspots is often different to that which is described above. In a public place where there are many networks, the clients often have to begin by choosing the network to which they wish to connect, and then identify themselves to the access point. Finally, in most cases, they need to enter authentication data. There are solutions in existence to simplify network selection and automatically make the connection whilst still ensuring good security, but these solutions are often restricted to a small number of networks and are very rarely available abroad.

Wi-Fi Passpoint technology fundamentally changes the use of public hotspots. It is a general solution which all hotspot operators can apply, which helps to overcome the limitations of the proprietary, non-interoperable solutions offered by current providers. A program, installed on certified devices, automatically manages the network connection, authentication, authorization and underlying security in a manner which is totally transparent to the user. This solution works with all Passpoint networks.

First and foremost, Passpoint takes care of the discovery and selection of a network. The terminal machines discover and automatically choose the network to which to connect on the basis of preferences determined by the users, policies defined by the operators and the availability of the network. These characteristics are based on the IEEE 802.11u standard.

The network connection takes place seamlessly – i.e. without the user having to do anything at all. It is no longer necessary for the terminal to be on an active list of access points to which the user agrees to connect. Nor is it necessary to enter account information into a browser. Passpoint uses a consistent interface, and the connection process is automatic. In addition, the choice of peripheral device can always be made automatically for multiple types of identification information. Passpoint uses SIM (Subscriber Identity Module) cards for authentication; these cards are also very widely used in modern-day cellphone networks. Passpoint also uses username/password combinations and certificates. No intervention on the part of the user is needed to establish a connection to a trusted network.

All Passpoint connections are secured with WPA2-Enterprise for authentication and connectivity, thus offering a level of security comparable to that of cellular networks. Passpoint improves upon WPA2-Business, adding functions capable of containing the known types of attack in deployments for public Wi-Fi networks.

Passpoint can also be used to instantaneously open accounts when the user does not have a subscription to a telecom operator. This process is also standardized and unified for the establishment a new user account at the access point, using the same account-creation method employed by all Wi-Fi Passpoint service providers.

Passpoint creates a global platform centered on four protocols based on EAP (Extensible Authentication Protocol). Standardized and supported by all manufacturers in the area of security, this protocol is very widely employed today. The authentication methods EAP-SIM, EAP-AKA and EAP-TLS enable mobile operators to use the same authentication solutions for cellular and Wi-Fi.

In addition to ease of use because of transparent authentication, Passpoint offers numerous advantages to service providers and users – in particular, the following:

– internet for electronic devices without a browser. Passpoint's authentication method does not require a browser, facilitating the use of electronic devices such as cameras, onboard devices in cars,

connected objects and, more generally, all “things” connected to the Internet;

– simplicity of connection and creation of new subscriptions, whether to attach them to an existing account or introduce new services. The automation of the authentication process and the security of the connection make Passpoint small-cell access an attractive solution for content providers and manufacturers of content-oriented terminals, such as e-readers. Occasional users of Wi-Fi can use prepaid subscriptions, with the model being similar to that used by mobile operators.

Service providers are increasingly eager to protect their subscribers’ paying content. In order to do so, they need to know who receives that content. Passpoint authentication enables service providers to verify identity, to have access to the subscribers’ rights and to offer premium-quality content for subscribers connected to their home network, a company network or public access points.

Passpoint hotspots offer service providers transparent roaming to other operators’ networks. To activate roaming, the service operators first need to establish mutual roaming agreements, covering access validation and charging. Roaming is based on a single protocol for network selection and user authentication in the hotspot.

Once roaming between two service providers is activated, Passpoint devices can connect automatically either to the network of their own service provider or to that of the other, using the same procedure. In all cases, the Passpoint client recognizes the access point as belonging to the list of available networks and establishes a connection automatically, as happens with roaming for cellphones.

The new functions are activated only if the access point and the client’s device are Passpoint-compatible. The Passpoint certification program run by the Wi-Fi Alliance ensures this compatibility. In addition, Passpoint supports connectivity with non-Passpoint devices.

Clients with valid identification data and using an appropriate EAP method can connect to Passpoint access points. Clients who do not have identification data appropriate for EAP methods have to use an

open system, based on authentication in a browser, where WPA2 may not be needed. In this case, security is not guaranteed.

Passpoint clients can connect to any existing Wi-Fi network. However, Wi-Fi clients will not necessarily be able to use Passpoint functions and services. For example, standard hotspots offer no security of connection, whereas Passpoint automatically implements WPA2 encryption.

Wi-Fi Certified Passpoint is a certification program offered by the Wi-Fi Alliance, aimed at both the access points and the clients' terminals. Just as for Wi-Fi equipment, it ensures interoperability of the devices and terminals.

In Figure 6.8, we illustrate the different types of connections that can be made by a mobile in a Passpoint environment.

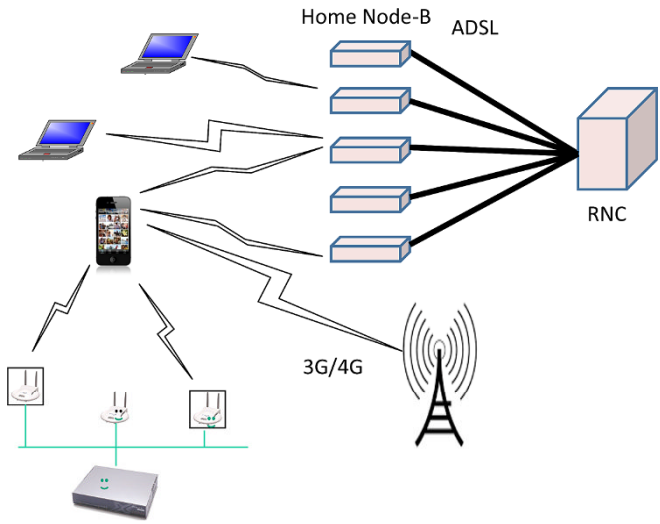



**Figure 6.8.** *The integration of 3G/4G/5G and Wi-Fi access*

In Figure 6.8, three types of connections are possible: a connection with the NodeB, which is the typical case of connection of a 3G/4G terminal; a connection with a hotspot which is, itself, connected to a Wi-Fi controller; and a connection with one or more HNBs

(Home Nodes-B), located in domestic environments. It would also be possible to add metrocells, which are similar to femtocells but situated in public spaces – e.g. a street – instead of in private areas. The connection choices are made by the Passpoint technology, which is based on a Cloudlet situated near to the access points. The power of the Cloudlets becomes essential in this scenario, because the number of actions increases with the possibility of handovers between two of the access points, establishing multi-paths if a user's data stream can be channeled through several access points simultaneously, and finally, managing multiple technologies by replacing certain Wi-Fi access points with connections such as Bluetooth, ZigBee or any other solutions available on the market.

## 6.12. Backhaul networks

Backhaul networks form the intermediary networks between the access networks to which the clients are connected and the core network. Such backhaul networks are becoming increasingly important, because small-cell technology requires a very large number of connections with all access points, instead of concentrating the connections on a small number of large antennas. It can be said that, up until now, backhaul networks have essentially consisted of links between DSLAMs and the core network, or between Nodes-B and the core network. Most typically, these networks use Gigabit Ethernet or MPLS technology.

The solution developing with small cells pertains to mesh networks, or networks of access points, in which the boxes are directly interconnected. For this purpose, mesh access points have two Wi-Fi ports: one to communicate with the neighboring access points, and the other to connect the clients. A client wishing to connect plugs into one access point, which transmits the packets to the neighboring access point, which in turn transmits them to its neighbor, and so on until they reach the core network, using an optical fiber or an *x*DSL modem. Typically, the access points are connected to one another by Wi-Fi, but other solutions are developing, with longer ranges, as we shall see.

Figure 6.9 shows a conventional configuration for a backhaul network. The small cells are connected to a "Small Cell Backhaul Aggregation" station, which harvests the data streams from the different cells and aggregates them, forwarding them on to the core network, either by fiber-optic or by wireless beams. The flows are directed to switches in the RAN (Radio Access Network), which are, themselves, connected to a switch in the core network. This link is often made using very-high-data-rate MPLS technology.

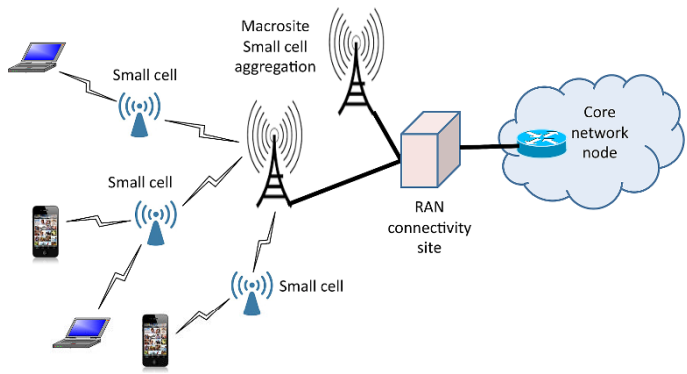



**Figure 6.9.** *A backhaul network*

The connections between the access points and the aggregators, or between the access points and the first RAN switch, increasingly frequently use wireless connections with a very high data rate, of several gigabits per second at high-frequency bands – e.g. the 60 GHz band. The major disadvantage to using these frequencies is that there must be a direct view and favorable weather conditions. Indeed, signals on these frequencies, which are sub-millimetric, are disturbed by drops of water and dust in the atmosphere. More specifically, the connection can use the 45 GHz, 57–64 GHz or 70–80 GHz bands, or even around 105 GHz. Seemingly, the most popular band is 60 GHz, for a variety of reasons: the bandwidth available, data rates countable in gigabits per second, low interference (because the connections are highly directional), low latency, small size of the components and antennas, and finally, a low production cost. The drawback to using this band is that two communicating antennas cannot be more than a kilometer apart, so as to avoid excessive losses.

## 6.13. Software radio and radio virtual machine

Undoubtedly cognitive radio represents one of the greatest revolutions of this new millennium. It facilitates much better use of the spectrum of radio frequencies, which is generally very poorly exploited. However, optimal use of the spectrum would be impossible without a certain number of other, parallel developments, which we shall now go on to discuss. We have already encountered this technology in TVWS (TV White Space) Wi-Fi in the IEEE 802.11af standard, pertaining to the use of unoccupied television bands.

Software radio, or SDR (Software-Defined Radio), defines a radio transmitter or receiver which is in software, rather than hardware, form. The implications of such technology are evident, because the characteristics of the radio interface can be modified infinitely without any technical problems at all; users will have the interface they need at a given time $t$, and a completely different interface at time $t + 1$. Obviously, a certain amount of hardware is needed to perform the computations of encoding, modulation and filtering, and for the transmission itself.

The computational power may be hosted on the device itself, or – as we have seen – in the mobile Cloud, in a local Cloudlet, a regional Cloud or even a central Cloud.

At present, progress in the area of this technology is being made very rapidly, because it is possible to condense all the computations for a large number of access points to a single machine. Nowadays, we are seeing the emergence of a particularly interesting, attractive solution: using a single antenna for all radio communications. The signal is sent to the appropriate virtual machine for processing. In other words, at certain times the antenna may receive Wi-Fi signals, and then at other times signals from a 3G communication, then 4G, then a TV signal, then Zigbee, and all of it is handled by virtual signal-processing machines.

The Cloud, once again, is important in these technological advances, and we can clearly see the amount of power and memory that are needed to support the necessary software machines. However,

sometimes, there is insufficient power to process the signals in real time if the transmission rate increases. This is the reason for the current drive toward concretization – i.e. the opposite of virtualization. Software is transformed into hardware, using reconfigurable microprocessors. Whenever a specific program is needed, a hardware version of it is created, which increases the processing speed 100-fold. As the process of transforming the hardware is, as yet, relatively slow, we generally need several reconfigurable processors in order to offer real-time applications: whilst one processor is transforming, the others are running.

In Figure 6.10, we have represented what can likely be expected from software technologies in the world of radio over the coming years. The objectives are plotted on the basis of the flexibility that software radio can offer and the intelligence offered by cognitive radio.

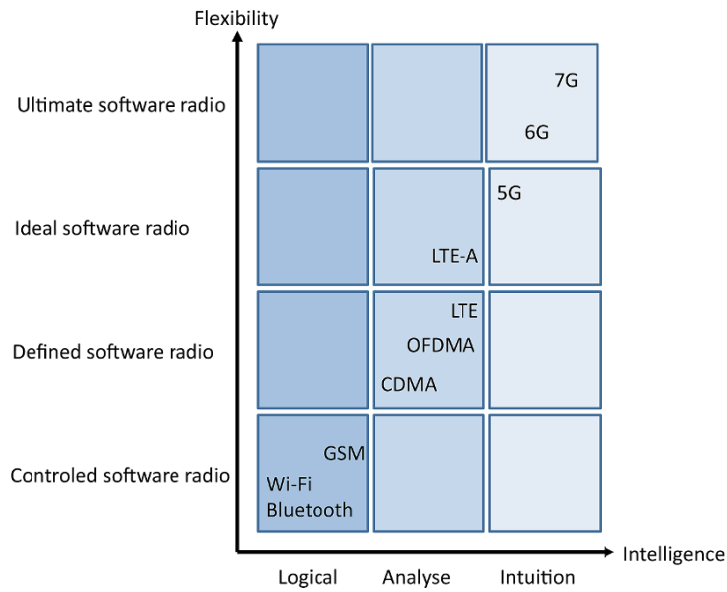



**Figure 6.10.** *The different advances made in the field of software radio*

Thanks to these technologies, it is predicted that data rates will increase by a factor of 1000 within the next ten years.

## 6.14. 5G

5G is 4G's successor. It is likely to be 2020 before it is standardized, and a few years after that before it begins being commercialized. The first major development which it offers regards the peak data rate, which should be between 1 and 10 Gbps. The main characteristic is the Internet of Things – "things" essentially being medical devices, household appliances or objects and sensors.

The characteristics of 5G contain numerous key terms, including:

– UWB (Ultra Wide Band): use of a very wide spectral band, which can be purchased from states or used in cognitive radio;

– smart antennas: antennas capable of operating with diverse encoding techniques, and also practically in any frequency;

– small cells: as we saw earlier, small cells help to exponentially increase data rates by reusing the available frequencies;

– software-based approach, using virtualization: this approach is widely used with this new generation of technology. In our coming discussion, we shall look at several examples of virtualization, but this technology is also encountered in at least three main techniques:

- software-defined radio,

- software-defined networking,

- seamless combining of wide-band networks;

– virtualization: virtualization also takes place in the devices themselves, such as:

- HLRs, VLRs, eNodeB, RANs,

- Cloud-RANs;

– multi-access, multihoming, multi-technology and multi-route, which are technologies that are developing in the form of products available since 2014;

– D2D (Device to Device), D2D2D, mesh networks, which represent simplified access techniques for ease of access, reduced energy consumption and reduced cost of connection.

Figure 6.11 represents several situations that can be handled by 5G. Starting on the left-hand side of the figure, we see multi-hop communications: the terminal is too far away from the NodeB, and the signals pass through intermediary machines in order to reach the antenna. This solution helps to limit the number of antennas whilst covering a large geographical area. Mesh networks or *ad hoc* networks provide this type of service, using algorithms to route the signals between the different terminals.

Next, we see D2D (Device to Device) – i.e. direct communication between two terminals without going through an antenna. This is a very eco-friendly solution, as it saves energy, minimizing the data rate of the system and thus decreasing electromagnetic pollution. Then, we find high-reliability solutions, whereby one path to reach the antenna can be replaced by another path almost instantaneously. Next, the figure shows connections to machines, for M2M (Machine-to-Machine) or M2H (Machine-to-Human) communication. The connection of things is also represented by medical devices, household appliances or other items. The figure also illustrates the connection to an intermediary relay to reach the main antenna. Indeed, as the energy expenditure depends on the square of the distance, it is very much to our advantage to use devices with short ranges.

Next, in the same figure, we see the problem of high density – e.g. the coverage of a stadium holding 100,000 people, with each spectator being able to enjoy a data rate of around 10 Mbps. The total data rate of 1 Tbps, in this scenario, could easily be delivered by an optical fiber, but the difficulty lies in the distribution of that capacity wirelessly. 4G and 5G technologies on their own are incapable of delivering such performances: we would need either extremely small cells or extremely high bandwidths.

Finally, Figure 6.11 shows a Cloud, which is one of the crucial elements – in fact, the very center – of 5G technology, in forms that have yet to be determined. However, we can confidently state that the

algorithms will be decoupled from the devices, and will run on Clouds of greater or less power, situated more or less near to the user.

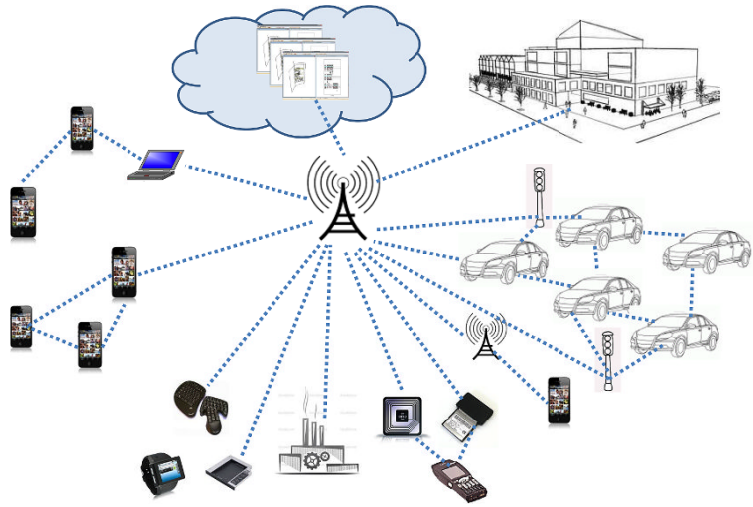


**Figure 6.11.** *The different 5G access solutions*

In summary, 5G networks should facilitate the integration of multiple wireless/mobile networks, which should allow for all classic applications and those of future society. This generation should also provide solutions to numerous problems; a few examples will be mentioned below. Most solutions will be heavily dependent on virtualization. The main problems are expressed by the following questions:

– how can we avoid having too great a number of access points?

– how can we avoid too cumbersome a cable infrastructure if numerous operators have developed small-cell infrastructures?

– how is it possible to ensure quality of service in these overlapping and interlocking networks?

– how can the backhaul networks ensure continuity between heavily-virtualized endpoints of the network and the core network, which is, itself, virtualized?

As stated many times, virtualization is one of the main solutions to the problems posed by fifth-generation technology. Looking first at the peripheral devices, connections can be gathered together on virtualized shared access points, so as to avoid having to deploy a physical access point for each small cell and each operator. The virtualization of a Wi-Fi access point is illustrated in Figure 6.12. The physical antenna is shared between all the virtual access points. The physical box which contains the virtual machines has a hypervisor, upon which the virtual machines are founded. Each virtual access point is in one of those virtual machines. The characteristics of the access points may be totally different from one another. Evidently, it is possible to replace the Wi-Fi access point with a 3G/4G base station which, if we employ virtualization, can actually become several base stations.

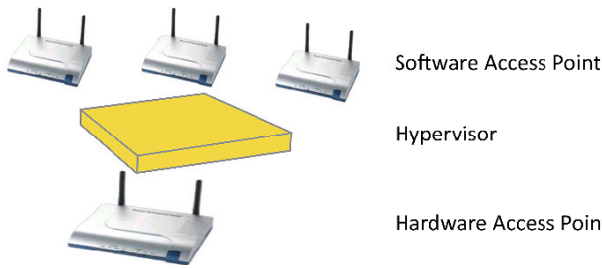



**Figure 6.12.** *Virtualization of a Wi-Fi access point*

Similarly, an HNB (Home NodeB) can be virtualized and play host to several operators without them having to compete for the infrastructure. Furthermore, the mobility offered by HLRs and VLR also weighs in favor of the virtualization of these devices, in order to share these servers between multiple operators. Obviously, the end goal is to lower prices significantly. Figure 6.13 illustrates the virtualization of two devices used for 5G: the virtualization of a Wi-Fi access point and that of an HLR.

The virtualization of a Wi-Fi access point leads to the creation of several cells which overlap one another, and are served one by one, taking turns depending on the division of the access point's resources. However, the packets passing through the same physical antenna can

only pass through one after another. This is completely different from IEEE 802.11ac technology, where each antenna has its own cell, which does not overlap the other cells, so that multiple communications can take place at the same time in different spaces. Similarly, in a virtualized HLR, the resources are used successively when it is the turn of each successive virtual machine to be active.

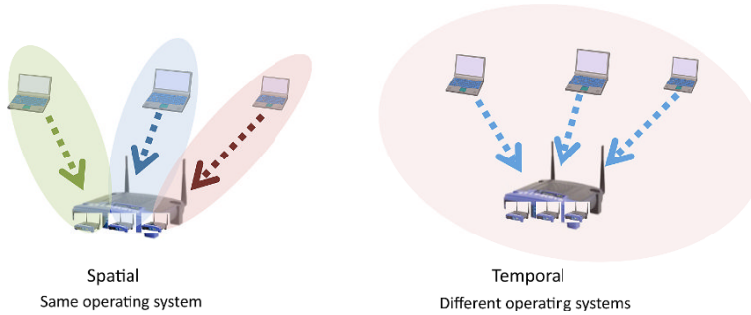



**Figure 6.13.** *Virtualization of 5G devices*

The virtualization of an HNB (Home NodeB) or an MNB (Metro NodeB) is illustrated in Figure 6.14: the box has a hypervisor which supports virtual HNBs. Each HNB or MNB uses its own technology, provided it is compatible with the common antenna and the resources of the node. In the case of a domicile or a shared property, this solution can be used to share a box between several operators. Additionally, each operator can perfectly well manage multiple software networks in the same box for different applications. The Achilles' heel of this technology is the antenna's capacity, which is shared between the different clients. The advent of 11ac, 11ad, 11af and 11ah technology should facilitate the introduction of these new devices, thanks to very high data rates. With virtualization, each user is given the illusion of having the desired capacity all to him/herself.

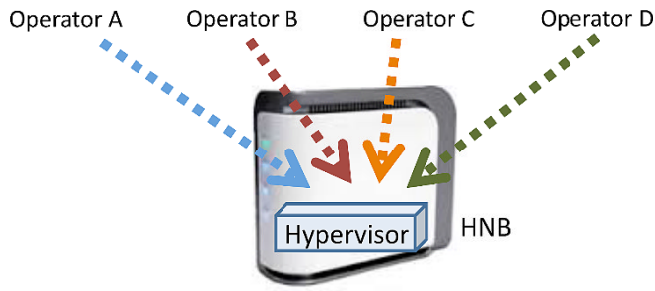



**Figure 6.14.** *Virtualization of an HNB*

Virtualization is found in various types of 5G networks, such as mesh networks. This scenario is illustrated in Figure 6.15. The machines which manage the mesh network have two interfaces: one interface with the users and the other with the other machines in that mesh network. These machines are virtualized with a hypervisor which supports mesh routers or switches, depending on the technology employed – e.g. using a TRILL protocol. It is possible for the machines to communicate by using Ethernet routing. Of course, there are IP routing possibilities which have been greatly developed by the MANET group at the IETF. One of the advantages of this solution is that certain networks can be allocated greater resources than they could have had if there had only been one network.

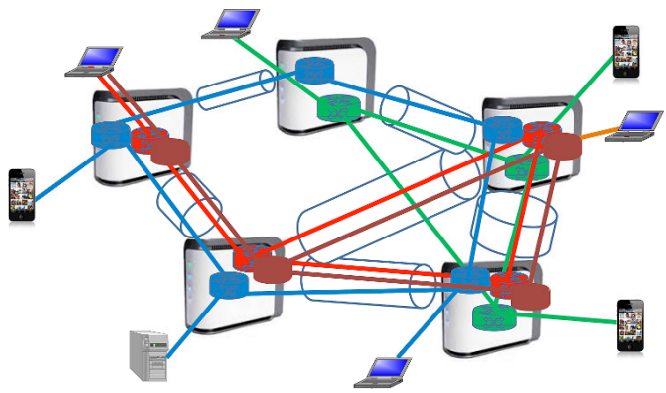


**Figure 6.15.** *Virtualization of a mesh access network. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

Look at the example of a VoIP network (Voice over IP): the software network assigned to it may contain far more resources than are actually needed by the VoIP flows. As the networks are isolated, the VoIP network is sure to receive at least the amount of resources that it really needs in order to deliver the desired level of QoS. Any resources which have not been used will be reallocated to other networks. Thus, we can determine a hierarchy of priority, in the knowledge that the network with least priority will have at least the resources assigned to it when the resources are apportioned to each network.

We can do exactly the same thing for the backhaul networks, which link the access networks to the core network. By virtualizing the backhaul network, we are able to deliver quality of service to the virtual networks which need it. In addition, if the access networks are virtualized, the backhaul networks must be virtualized as well, so that the classes of service are compatible. Figure 6.16 shows a backhaul network which can be virtualized by nodes supporting hypervisors and hosting appropriate virtual machines.

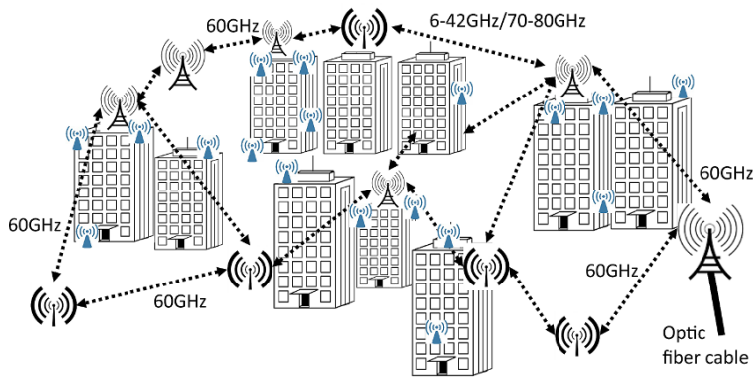



**Figure 6.16.** *Virtualization of a backhaul network*

## 6.15. C-RAN

C-RAN, or Cloud-Radio Access Network, is a system put forward in 2010 by China Telecom, with the aim of profoundly changing the

access and backhaul networks between the core network and the peripheral devices. To begin with, the architecture is based on the Cloud: all the control algorithms are handled by datacenters. This architectural arrangement could have been reached simply by continuing to use the conventional access networks connecting clients to the Nodes-B. Nonetheless, the proposition is fairly revolutionary, in that the access network is eliminated completely, and replaced with a very simple solution whereby the radio signal is sent directly to the datacenter. The radio signal is transferred to the Cloud as it is, without being embedded in a packet. The signal processing takes place in the Cloud. The original idea of C-Cloud is shown in Figure 6.17. The terminals are connected to the antennas, which re-transmit the signal to the central cloud, where it is computed. Once it is received by the antenna, the signal is sent back over an optical fiber, using a technique known as RoF (Radio over Fiber). The major advantage of RoF technology is that the same antenna can be used to handle very different signals, such as 3G, 4G, 5G or Wi-Fi. It is the Cloud which unscrambles the signals and determines their characteristics in order to be able to decode.

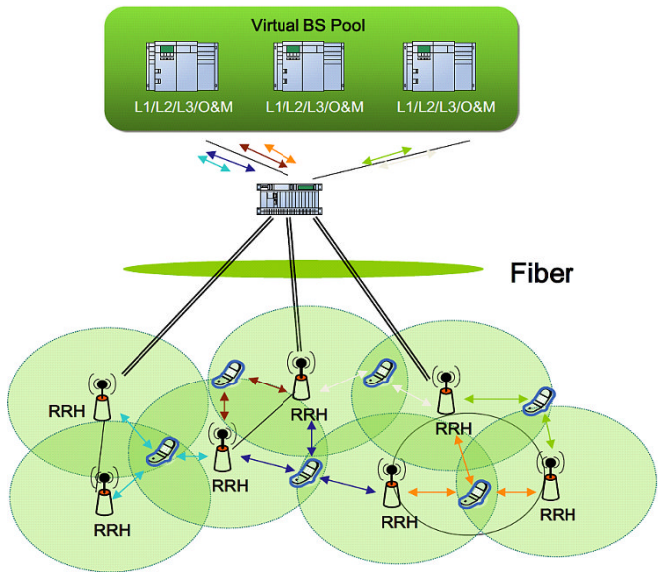



**Figure 6.17.** *The fully-centralized Cloud-RAN architecture*

We can also cite other advantages, such as the low attenuation of the signal over the fiber, so the datacenter can be far away from the source. The cost of connection is also going down, thanks to multiplexing on the same antenna, but also on the same optical fiber. The complexity of the access network is eliminated, and the datacenter handles this complexity with appropriate virtual machines. This technology is found in the cabling solutions used at certain large stadia or shopping centers.

The main disadvantage stems from the use of FTTA (Fiber to the Antenna) which is necessary for the communication. If the building containing the Wi-Fi access points is not cabled, then installing an optical cable can often be extremely costly. The optical fiber is connected to the datacenter by a BBU (BaseBand Unit) and, on the other end, is connected to the physical antenna by an RRH (Remote Radio Head).

The standardization currently under way for NFV (Network Function Virtualization) includes C-RAN. Functions in the NFV environment are defined to decouple the work carried out on the node, which was previously done by the antenna, and move it to a datacenter.

Whilst the C-RAN architecture is an interesting option for countries wishing to develop a new infrastructure, it is much less attractive to countries which already have a local-loop infrastructure for RAN. In such cases, the proposition is to keep the existing structure and add Cloudlets near to the antennas to perform computation locally. The computation pertains to the signal but also to the signaling, with algorithms to manage handover or the choice of connection to be favored when multiple options are available. This architecture is illustrated in Figure 6.18.

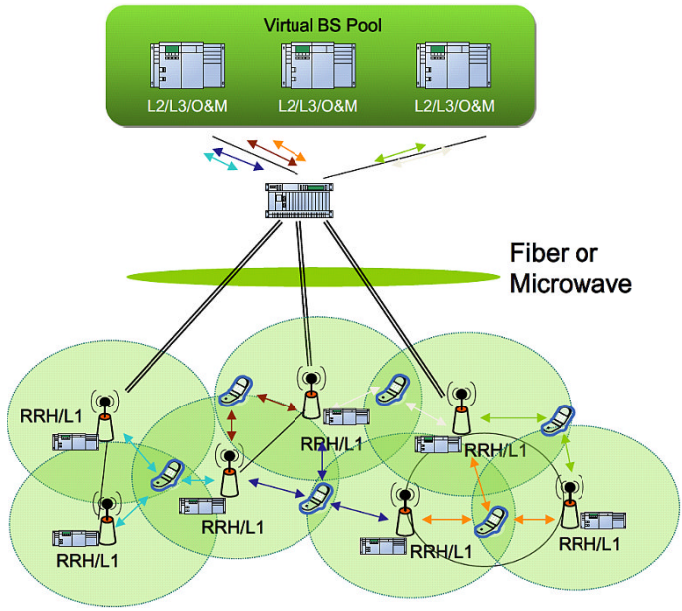



**Figure 6.18.** *The partially-distributed Cloud-RAN architecture*

## 6.16. The Internet of Things

The Internet of Things was born with the idea of connecting wireless and hardwired sensors, found in the home, at work and more or less everywhere in life today, to the Internet. One of the most widely used sensors is RFID (Radio-Frequency Identification), or electronic tags. The Internet of Things enables us to connect anything and everything that is connectable, from varied objects to smart dust. The concept is simple, but the problems are numerous, because in general, the "things" in question are not sufficiently sophisticated to handle the communications and computations associated with the applications.

We shall begin by examining sensor networks, followed by RFID networks and the NFC (Near-Field Communication) interface – which is increasingly being used in the connection of "things" to the Internet – and HIP (Host Identity Protocol), which could become the standard for interconnection gateways between "things"

and the Internet. Finally, we shall touch on a few ideas about one of the most rapidly developing types of networks: medical sensor networks, which could potentially affect seven billion people, and constitute a truly enormous potential market.

## 6.17. Sensor networks

A sensor network is defined as a set of interconnected sensors, with each sensor having a transmitter-receiver. Sensor networks form a new generation of networks with specific properties, which do not fit into the framework of conventional architectures. However, the dawn of the Internet of Things has altered our view of sensor networks, which may, of course, still constitute a closed system, but also connect to the Internet.

The miniaturization of the sensors poses problems in terms of communication and energy resources. The sensor needs to be sufficiently intelligent to gather the required information and transmit it correctly. In addition, the sensor's processor must not be used too intensively, so as to consume as little energy as possible. Thus, it must incorporate reactive elements, rather than cognitive ones. Finally, in order to ensure a good data rate, the range of the transmitter-receivers must necessarily be small – around ten meters. Therefore, the establishment of a sensor network poses problems of routing, error control and power management.

Very special sensors are involved in what we call "smart dust". These dust particles, which are practically invisible, have a radio device in addition to performing the computations relating to the internal sensor.

From the standpoint of communication, the environment of IP is too intensive, leading to an excessive data rate and overconsumption. Solutions derived from terrestrial networks, or industrial real-time networks, offer a better compromise between efficiency and power consumed. As there may be hundreds of such sensors per square meter, IPv6 routing represents one obvious solution to deal with the problem of addresses. However, the IP environment is cumbersome,

and it is often impossible, in these sensor networks, to use a TCP/IP or even UDP/IP protocol stack.

For the moment, security and QoS problems are taking a back seat to energy consumption problems. In any case, a great deal of research is currently being done to render sensor networks efficient and robust.

The main radio standards are ZigBee, WiBree and 6LowPAN. WiBree is a very low-consumption technology with a range of 10 m and a data rate of 1 Mbps. This solution was developed by Nokia to compete with both ZigBee and Bluetooth. However, it was integrated with Bluetooth in 2009, with the resulting product being called Bluetooth LE (Low Energy).

6LowPAN networks (IPv6 over Low power Wireless Personal Area Networks) were proposed by a working group at the IETF. The objective, obviously, is to facilitate continuity of IP on non-powerful machines with limited electrical power.

The use of the IPv6 standard to obtain a very large number of addresses for immense sensor networks poses a problem. The 16 bytes occupied by the transmitter's address, added to the 16 bytes of the receiver address, plus the obligatory fields, lead to poor usage of the radio link to transport the supervision data. This could become genuinely problematic in light of how little energy the sensor has. ZigBee, on the other hand, limits the length of its frame to 127 bytes, which may also cause problems if the message supplied by a sensor for transmission is long.

Sensors form *ad hoc* networks, and they need a routing protocol. The use of a protocol such as IEEE 802.11s, in combination with IPv6 addresses, would be catastrophic for the sensors' battery life. For this reason, current propositions are much simpler, with protocols such as LOAD (6LowPAN *Ad hoc* On-Demand Distance Vector Routing Protocol), a simplified version of AODV, DyMO-Low (Dynamic MANET On-demand for 6LowPAN), a simplification of DyMO, from the MANET working group, and Hi-Low (Hierarchical Routing over 6LowPAN). These different protocols stem from propositions by the IETF, and therefore the standardization of *ad hoc* networks, but they do not include all the potential options.

Another important characteristic of the protocols used in sensor networks is service discovery, which should enable the automatic establishment of a connection. The IETF also plays an important role in this domain, having published several proposed solutions – one of which is sensor-oriented: LowPAN Neighbor Discovery Extension. This protocol is a reduction of the Neighbor Discovery standard, which pertains to all energy-consuming elements, including broadcasts and multicast management.

A very special example of a sensor network is offered by smart dust, the aim of which is to develop nanotechnology sensors and to connect them to one another by an *ad hoc* network. Smart dust fits into a space smaller than a cubic millimeter – hence the name. Each grain of this dust contains the components necessary to constitute a communicative computer: a processor, memory, radio, battery, etc.

Here, the main problem is the saving of energy while performing sensor functions. In particular, the network part must carry out communications using very little energy indeed. Thus, Berkeley University has designed a specific operating system and protocols, known as TinyOS and Tiny protocol. TinyOS is written in simplified C language, called nesC, which is a sort of dialect designed to optimize memory usage.

## 6.18. RFID

RFID (Radio-Frequency Identification), or radio-identification, was introduced for the purpose of identifying objects, so it is also known as electronic tagging.

Electronic tags are scanned by a reader, which is able to recover the identification data. The tags are used in numerous applications, from the tracking of animals to security tags in stores.

There are two main types of electronic tags: passive and active tags. Passive tags have no energy source. They are activated by a reader which supplies a sufficiently strong electromagnetic field to generate an electrical current, facilitating the radio-wave transmission

of the binary elements stored in an EEPROM memory, constituting the RFID identification. A passive tag is illustrated in Figure 6.19. The antenna must be built in such a way as to be able to receive the electromagnetic field from the reader and transmit its identity.

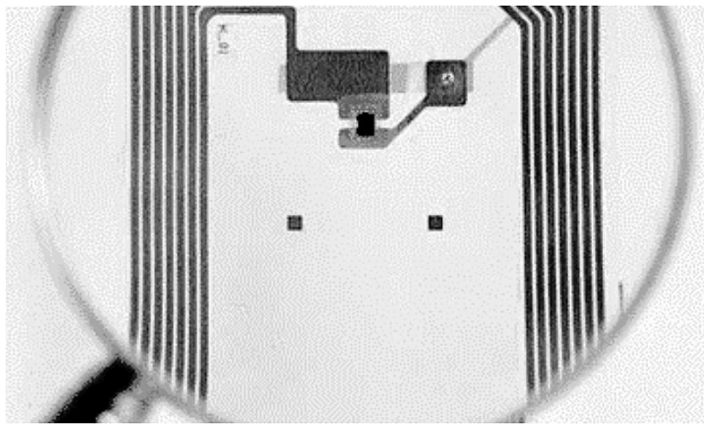


**Figure 6.19.** *An RFID*

A passive RFID may be very small. As the equipment requirements are limited, sizes of around a tenth of a millimeter are sufficient. The price of an electronic tag depends on the number of elements manufactured in one batch. It is possible to go as low as 5 euro cents.

Active electronic tags have a source of electrical supply within the component. The first, extremely significant, advantage of such tags is the quality of the transmission. A session can be established between the reader and the RFID so that the signal is automatically retransmitted in case of a problem. Another advantage is transmission with a range of several meters between the RFID and the reader, instead of only a few centimeters. One disadvantage could be the battery life. However, with typical usage of a few readings per day, it is possible to achieve battery life of over ten years.

The active RFID may have a larger memory for storing attributes associated with the value of the ID.

An active RFID is shown in Figure 6.20.

**Figure 6.20.** *Active RFID*

One very well-known application of RFIDs is the electronic passport. The e-passport is defined in a text published by the ICAO (International Civil Aviation Organization). The chip in the e-passport contains the same data that are printed on the passport itself, along with a digitized photo of the passport holder.

Transport documentation (tickets, etc.) is a second application for RFIDs. For example, the tickets sold on the Paris Metro contain an electronic tag which memorizes a set of data including the date and place of purchase. More sophisticated solutions are used on public transport in Seoul, where the tag is active and contains money, so that the same ticket can be used multiple times.

Toll barriers on the roads also use this active RFID solution, with ranges of a few meters. The active tag is able to subtract the cost of the toll from the amount of money stored in the memory. Similarly, toll barriers for the ski lifts in many resorts in France use active RFIDs.

Another fairly immediate application is the tracking of cars, to detect stolen cars when they pass near a RFID reader.

Finally, two of the best-known applications of RFID are for inventory and for purchases in stores. Inventories can be taken more often, and with fewer errors. Similarly, items put into a shopping cart

can be recorded by the reader, which greatly simplifies the process of payment for those items in a shop.

RFID transmission frequencies are determined by local or regional standardization bodies. The main ones are indicated in the following table:

| Frequency for RFID | Comment |
|---|---|
| 125KHz (LF) | First solution to offer a relatively large range for passive RFIDs |
| 13.56MHz (HF) | One of the standardized frequencies very widely used for passive RFIDs |
| 400 MHz | A number of specific uses, such as the detection of stolen vehicles |
| 865-868MHz (UHF) | Frequency band standardized in Europe for intensive RFID use |
| 902-928MHz (UHF) | Frequency band standardized for North America |
| 2.4-2.4835GHz | SM open band in which numerous RFID applications are likely to develop |

**Table 6.1.** *RFID transmission frequencies*

## 6.19. EPCglobal

The purpose of RFIDs is to give the identity of the objects to which they are attached. This identification system has been standardized by the consortium EPCglobal. Two generations are available: EPC Gen1 and EPC Gen2. We shall focus particularly on this second generation, released in mid-2006, which has become an industrial standard.

EPC Gen2 is the acronym for "EPCglobal UHF Class1 Generation 2". Version 1.1 of this specification was released in May 2006. It handles the protocol between the reader, on the one hand, and the RFID and the identity, on the other. The object of the protocol is to read, write and eliminate an RFID, so that readers sold by all manufacturers are interchangeable.

The reading procedure is defined using a timeslot-based system with an anti-collision system. Specifically, a reader can

simultaneously trigger a large number of RFIDs, but the simultaneous reading of different tags would lead to collisions. Signaling is used to determine the frequency, the coding used (between DSB-ASK, SSB-ASK and PR-ASK) and the data rate of the channel. The anti-collision system means that, whenever tags are read simultaneously, only half of the objects that have been able to transmit are allowed to transmit again the next time. After a certain number of collisions, only one RFID is able to successfully transmit. The algorithm is designed in such a way that each RFID then takes turns to transmit. The reading speed may be up to 640 Kbps.

The identity of the object is determined by the EPCglobal Electronic Product Code. Gen1 uses 96 bits, whilst in Gen2 the code is 256 bits in length. An example of this ID for Gen1 is illustrated in Figure 6.21. Figure 6.22 shows the ID fields in Gen2. This solution is much more complex, because it uses intermediary filters to determine the lengths of the subsequent fields. Note that the serial number grows from 36 to 140 bits, which means that, for a given article, the value never has to return to 0.

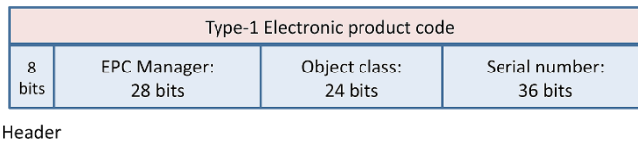



**Figure 6.21.** *Structure of GEN1 Electronic Product Code*

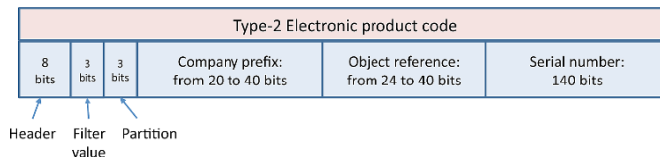



**Figure 6.22.** *The structure of GEN2 Electronic Product Code*

## 6.20. Security of RFID

Security is a thorny issue in the world of RFID. Indeed, a passive RFID device can be easily read by a third-party reader. In addition,

individuals' privacy may be violated by monitoring and tracing of everything relating to those individuals.

There are solutions available, such as encoding the ID in the tag or changing the value of the tag every time it is read. These solutions rely on middleware which is able to interpret the values of the tags or keep track of the changing values.

Active tags can establish an authentication session, facilitating an exchange with the reader, which then acts as an authentication server. In this case, we need to use an electronic circuit in the tag, capable of encrypting a text transmitted by the authentication server. However, the encryption keys used are so short that there is a not-insignificant danger of them being broken after a series of authentication attempts. Numerous propositions have been made, using the anti-collision algorithm, which serializes the reading of the tags, for authentication.

## 6.21. Mifare

Mifare is the most widely-used of contactless cards, with four billion in circulation throughout the world. The name comes from the company with developed it – Mikron; the full name of the card is "Mikron FARE". Mikron was bought out by Philips, which ceded ownership of this technology to its subsidiary: NXP. There are two different sub-types of Mifare cards: Mifare Classic, which uses only a certain portion of the Mifare command set, and Mifare from NXP, which implements the entire range of commands.

The cards communicate with a reader, which must be at a distance of less than 3 centimeters. This provides a certain amount of security for the communication, which is highly localized. However, attacks have been carried out, using very specific readers placed a few meters away. If we require security in a communication, the best thing to do is to encrypt the data for transmission.

These cards are not very powerful at all, so their production costs are extremely low. Mifare Classic cards have a 4- or 7-byte serial number and a storage memory between 512 bytes and 4 Kbytes. The

cheapest version is the Mifare Ultralight, which usually serves as a disposable ticket.

Mifare T=CL cards use secure elements of the same type as contact chip cards, and essentially provide the same type of service.

The most highly developed card is Mifare DESfire, which has a larger memory and an advanced operating system, as well as AES and 3DES encryption protocols.

Mifare cards serve numerous applications. The most common include: the reading of a serial number, to trigger a particular action if that number is accepted by the reader; the reading of an ID memorized on the card; or data storage. For the first application, we use the card's serial number, which is encoded on 4 or 7 bytes. In the second case, the ID must be entered into the card, which may encrypt it, provided it has a secure element capable of carrying out cryptographic computations.

## 6.22. NFC (Near-Field Comunication)

The NFC standard is a special case of RFID communications. It facilitates communication between an RFID and a reader over a distance of up to ten centimeters. This is contactless communication, with, for instance, a chip card or a secure microcontroller.

There are numerous and varied applications of this type. The best known is payment over a mobile phone. With his/her mobile, a client can top up his/her account with a simple phone call or low-data-rate data transfer to a server. Once the account is topped up, the mobile serves as a key to pay for a whole range of services. For example, to buy a subway ticket, we need only hold the mobile near the reader for radio communication to validate the purchase of the ticket.

NFC data rates are fairly low: 106, 212, 424 and 848 Kbps. The frequency range is around 13.56 MHz. The standards in this field were issued by the ISO/IEC. The NFC Forum was set up by Philips and Sony.

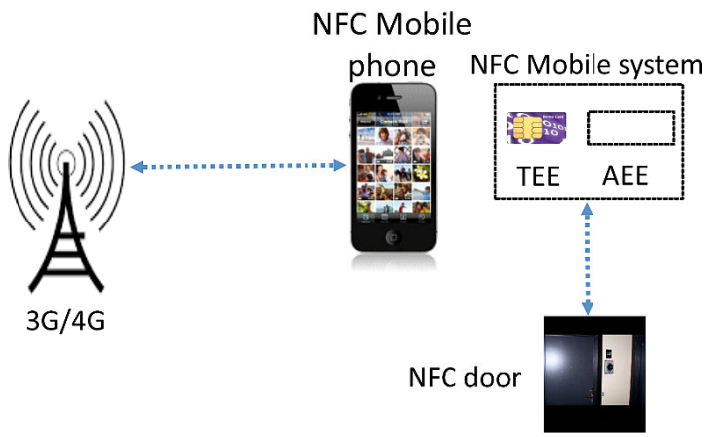
The security of the communication is assured because of the very short distance between the transmitter and the receiver and the presence of the chip card. Nevertheless, in spite of the maximum distance of 3 centimeters, in 2012, attempts were made to spy on NFC remotely. Therefore, it is advisable to encrypt the communication.

We shall now look at an example of the use of NFC, with mobile keys.

## 6.23. Mobile keys

Car keys, house keys or hotel keys could soon be included in smartphones. One of the advantages of mobile keys is that they can quite easily be transmitted to a friend waiting outside your door, or sent to customers by car rental companies. We shall describe the way in which this application works by taking the example of opening a hotel door with a smartphone.

Mobile keys are stored in a secure server or in a datacenter in a Cloud. The client has a smartphone, containing a secure zone to store the key. The door to the room has an NFC lock. The environment of this example is illustrated in Figure 6.23.




(AEE: Application Execution Environment, TEE: Trusted Execution Environment)

**Figure 6.23.** *The environment of a mobile key*

To begin with, the room is reserved using the mobile terminal. The smartphone must contain a secure element, i.e. a placement which cannot easily be reached from outside. This secure element may be a variety of different types, depending on the terminal (smartphone, tablet, laptop, etc.). It could be a SIM card from the operator, but in this case we also need the operator's authorization to use it. It may also be a so-called "embedded SIM", which is the equivalent of a chip card, but inserted by the manufacturer that made the smartphone. Most of the time, it is an NXP component which possesses much more than 50% of the market. It may also be a memory card (SD card) which is embedded in the mobile phone. Finally, as indicated in Figure 6.23, it may be an intermediary chip card (a loyalty card from the hotel where the client wishes to stay) that communicates with the mobile through the NFC interface. The mobile, in this case, is seen as a modem which facilitates communication between the external chip card and the key server. The key server is hosted in the Cloud. The door itself is NFC, and can be opened by a digital mobile key situated on the smartphone or on the external NFC card.

When the client arrives in the vicinity of the hotel, it is detected by the smartphone's GPS. The phone then transfers the key to the room into the secure zone of its security apparatus or directly to the external NFC chip card. We then merely need to bring the smartphone or the external card near to the NFC interface of the card to open the door.

## 6.24. NFC contactless payment

At the start of 2014, the number of contactless bank cards was 20% of the total number of bank cards. Similarly, 20% of smartphones had an NFC chip. Before the dawn of NFC, other payment systems had been tried, such as SMS payment, a direct-payment model and a solution using WAP. Over the past few years, payment using NFC cards has expanded hugely, and numerous variants have been defined.

In the context of NFC, the mobile must be brought to within a few centimeters of the payment terminal. Two major axes for this have been established: prepayment and direct payment. In the first case, the terminal is seen as an electronic wallet: with each transaction, money

is taken from the wallet. The payment for a very small maximum sum can be assimilated to this first axis. The second case requires more security, because payment is taken directly from the user's account, with the sums involved generally being much higher.

There are four possible models of mobile payment:

– *Operator-Centric Model*: the mobile operator alone deploys the mobile payment services. To do so, it needs to provide a mobile wallet that is independent of the user's account. This scenario is pretty rare, because the operators are not linked to the major international payment networks. The mobile network operator must manage the interface with the bank network in order to be able to provide advanced mobile payment services. Pilots using this model have been launched in developing countries, but they offer only very partial coverage of the numerous use cases of mobile payment services. In general, the payments are limited to transfers or to modest sums;

– *Bank-Centric Model*: the bank runs mobile-payment applications associated with client machines, and asks retailers to have the necessary equipment to effect the sales. The mobile network operators are responsible for the transport network, with the quality of service and the security needed for the payment functions;

– *Collaboration Model*: this model involves a collaboration between the banks, the mobile operators and a trusted third party;

– *Peer-to-Peer Model*: the mobile payment service provider acts independently of the financial institutions and of the mobile network operators.

Google, PayPal, GlobalPay and GoPago use a Cloud-based approach to make mobile payments. This approach places the mobile payment provider at the heart of the operation, which involves two distinct steps. First of all, a payment method associated with the Cloud is selected, and the payment is authorized via NFC. At this step, the payment provider automatically covers the cost of the purchase. In the second step, another transaction is necessary: the user's bank must reimburse the payment provider for that payment and any fees associated therewith.

## 6.25. HIP (Host Identity Protocol)

The IETF has developed various mechanisms to integrate things with the Internet. HIP is an identification technology which is part of this set of measures. This protocol is standardized by the IETF in RFC 4423 and the series RFC 5201 to 5207.

In the world of IP, there are two main naming conventions: IP addresses and the domain-name system. The purpose of HIP is to separate the endpoint ID and the localization of IP addresses. This approach is of the same type as LISP, described in Chapter 4.

For this purpose, HIP introduces a namespace based on a PKI infrastructure. This space enables us to manage multihoming securely. The IP addresses are replaced by encoded Host Identities (HIs).

## 6.26. The Internet of Things in the medical domain

The use of the Internet in the medical domain is on the rise, with 7 billion people to monitor, diagnose and care for. The long-term idea is to be able to continuously monitor individuals with a view to advising them or alerting them to any problems detected. *A priori*, in the future, everybody will be able to have uninterrupted connection to a Cloud, which will analyze the data provided by sensors both inside the body and on the surface. The power of the Cloud will be such that we will be able to analyze, compare and make diagnoses on the basis of this stored dataset. Essential characteristics in this field are security, "privacy" – i.e. the private nature of the data – and the very low energy consumption of the sensors, which must be able to operate for months on end with no need to change the batteries. This environment is being established bit by bit, and we shall now go on to describe the main advances from the networking standpoint.

The first element that we shall discuss is the sensor network adapted for use with the human body, called a BAN (Body Area Network). An example of a BAN is shown in Figure 6.24.

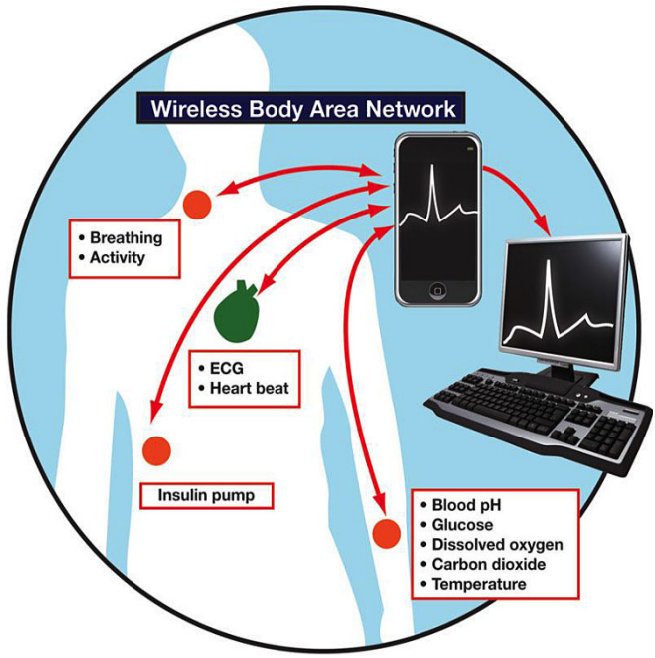



**Figure 6.24.** *A BAN (Body Area Network)*

It is the working group IEEE 802.15.6 which takes care of its standardization. The main standards issued by this working group cover multiple frequencies and multiple technologies:

– 402-405MHz for In-In-Out communications – i.e. inside the body and from the inside to the surface of the body;

– Ultra Wideband (UWB), which operates on the 2.4 GHz and 3.1-10.6 GHz bands;

– between 15 and 50 MHz for communications inside the body;

– HBC (Human Body Communications), which operates at 13.5, 400, 600 and 900 MHz for communications on the surface of the body. This communication uses the electrical conductivity of the human body. The intensity of the electrical current, therefore, is extremely low – around a nano-ampere.

To interconnect BAN networks with the Cloud, a number of solutions are envisaged: using the LTE network, or 4G, or a new generation of Wi-Fi network. In the United States, the FCC is working to develop a specific band for medical Wi-Fi, with much stricter security standards than those in force in Wi-Fi today. The bandwidth is 40 MHz, with a very low-consumption technology. The band for such external connections would be between 2360 and 2400 MHz – i.e. just below the current Wi-Fi band.

Other research, which is progressing well, regards the use of integrated sensors in "smart dust" capable of circulating in the blood. Analyses are regularly transmitted to external sensors which, in turn, transmit to the Cloud.

## 6.27. The Internet of Things in the home

The home is a place where we find a large number of very diverse sensors from different domains: ICT, telecommunications, medicine and electronics. For instance, we find Internet beds which have sensors that are able to detect all movements and speech during the night to aid in the diagnosis of sleep problems. Doors, household appliances, lightbulbs, security systems, etc., are, themselves, filled with increasing numbers of varied sensors. Another example is given by presence sensors, which detect whether or not a person is in a room, with a view to automatically turning the lights on or off. Overall, we expect to find around 100 sensors in every home by around 2020.

There is a major struggle going on between household appliance manufacturers, to determine which type of connection technology all those sensors will use in the home. The base network uses Ethernet, cable or Wi-Fi. Hence, most devices can be connected directly using an Ethernet port or a Wi-Fi card. Bluetooth, 6 LowPAN, ZigBee and NFC are also in the running. A communications controller should be able to personalize all the connections and ensure they have adequate security, good availability, reasonable energy consumption and as low a cost as possible.

The issue stems from the new generation of Home Gateway, which raises the question of management of IP in the terminal machine. For television sets, telephones and other fairly important machines, the presence of an IP component poses no problem: this is what we find in IPTV or IP telephony. On the other hand, for a very cheap sensor or device, it is either impossible or too costly to introduce an IP component and have an intermediary machine such as a controller. The controller can also be integrated into the Home Gateway. HIP, described a little earlier, can be used to perform communications, firstly with the sensor, and secondly with the Home Gateway.

In conclusion, the home contains a large number of "things" which it is possible to connect. Therefore, the Home Gateway is of prime importance, as it allows us – perhaps with a controller – to manage not only the communications, but also the security and quality of service of the "things" in the home.

## 6.28. Conclusion

5G is an environment that brings together numerous networks with different architectures and different protocols. This combination of technologies to form 5G is made possible by the arrival of the Cloud, Cloudlets, SDN (Software-Defined Networking) and SDR (Software-Defined Radio). Its design ranges from D2D-type solutions to the connection of billions of more or less intelligent things. Virtualization plays an important role because, almost everywhere, there is a decoupling between the physical device and the control device, which is usually situated in a datacenter specialized for the application.

The Internet of Things is becoming a reality thanks to RFID and sensors. However, we are still a long way from being able to completely integrate all these small devices, because they are not powerful enough to support TCP/IP, and yet they are rather energy-hungry. In order to achieve this goal, we need to use very specific gateways, which can include the connected things on the one hand, and the Internet on the other. HIP could become the fundamental standard of this connection solution.

# 7

# Security

Security is of prime importance for the networks of today, and will become even more crucial in tomorrow's world. With the Internet, it is necessary to constantly be inventing new procedures to ensure the non-disclosure of numerous elements, such as the location, the names of the parties being authenticated, the semantics of the messages, private information, etc. Of the various possible solutions, we shall discuss two in detail: the use of secure elements and the Cloud of security. Whilst the first solution has been in use for a long time, the second is a new paradigm which is becoming increasingly widespread.

Security in the world of networking is a paradigm which does not have a simple solution, besides making improvements to the existing algorithms, of which there are already a very great number, with a view to dealing with new attacks. Yet this chapter discusses a new solution in the world of security: the Cloud of security – i.e. a Cloud whose purpose is to secure the data and networks in the world of operators, companies and the general public. An initial diagram of a Cloud of security is shown in Figure 7.1. The Cloud of security contains numerous virtual machines for security such as authentication servers, authorization servers, identity management servers but also firewalls and even very specific firewalls corresponding to a particular application. We also sometimes find secure element servers which may contain thousands of SIM cards or HSMs (Hardware Security Modules).

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

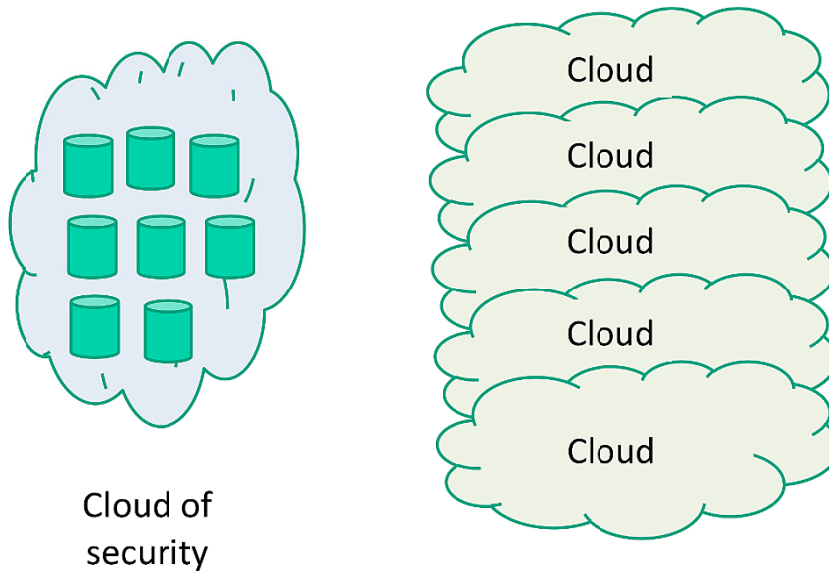



**Figure 7.1.** *A Cloud of security*

We shall now describe three examples of servers hosted in the Cloud of security, beginning with DPI (Deep Packet Inspection). The objective of DPI is to determine the applications transported in the flows passing through a network. In order to do so, DPI scrutinizes a flow and looks for applicative signatures which can, in general, be found by examining the grammar of an application. Every Internet application has its own signature. DPI examines the series of bits and determines the signatures. The advantage of this solution, as compared to examining the port number, is that there is a verification of all the bits passing through, rather than only the headers of the frames and packets. Indeed, attackers encapsulate their attack in messages of a known type which can easily pass through conventional firewalls. However, to detect signatures in a flow coming in at high speed is particularly complex and requires top-of-the-range – and therefore expensive – hardware. With the deportation to a Cloud of the function of examining the bits in a flow to find signatures, the cost of DPI can be greatly lowered. Thus, we deport the function of determination of the flows to a powerful datacenter. The cost is often the cost of transporting the flow which needs to be sent to the datacenter. Various

solutions have imposed themselves on the market, depending on the requirements. For example, only the message headers can be fed back, which dramatically reduces the flows to be examined, but brings with it the risk of missing encapsulated bits.

The second example is that of firewalls. Once again, the world of Cloud computing fundamentally changes these modules, deporting their software to datacenters with numerous advantages. The first advantage is to have specialized virtual firewalls for each application. With DPI, we detect the nature of the application and send the flow in question to the corresponding firewall. The firewall has processing capability to examine all details of the flow. The low-powered firewall dealing with all flows is replaced by a set of very powerful, specialized firewalls. The disadvantage relates to the flows, which need to be sent to the specialized firewall and must then be returned to the company, although there is the possible advantage of being able to hide firewalls in order to prevent denial-of-service attacks on them.

The third example is a secure element server which, as we shall see in this chapter, can be used to secure access to sensitive services such as mobile payment. These secure-element servers can contain thousands, or even millions, of secure elements such as smartcards, which may be reached by secure channels requiring high-security services.

Also, for inclusion in the Cloud of security, we could cite numerous servers such as authentication servers, identity management servers, encoding servers (although these have the peculiarity of having to be very near to the user), intrusion detection servers, etc.

## 7.1. Secure element

Strictly speaking, this chapter does not discuss security in the conventional sense at all, but rather a new generation which has gradually been being established over the past few years. This new generation uses secure elements as a basis. Indeed, high security cannot be satisfied simply by software, which can always be broken by a memory dump and good knowledge of the position of the keys

and other management systems. Secure elements are found in different forms, but the most commonplace today is the smartcard. Thus, we shall begin by describing smartcards, as the most classic secure element today.

Figure 7.2 shows a smartcard containing all the physical elements of an ordinary computer: a microprocessor, ROM, RAM, a persistent memory, generally EEPROM, a communication bus and an input/output. Until new smartcard embedded in a USB key were released, the communication channel represented a weak point, but this is no longer the case.

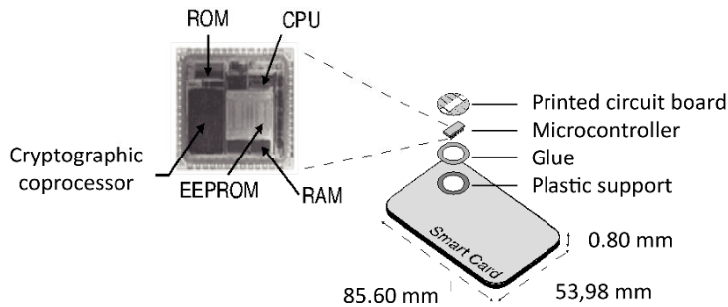



**Figure 7.2.** *Hardware architecture of the smartcard*

At the heart of most smartcards is an 8-bit microprocessor with computational power of around 1-3 MIPS (million instructions per second), at a frequency of 3.5 MHz. This type of microprocessor takes 17 ms to execute the encoding algorithm DES (Data Encryption Standard), for example. It should be noted that the power of smartcard processors is increasing at such a rate that new security cards will be able to support much more complex algorithms.

32-bit architectures based on RISC processors with 1 million transistors constitute the new generation of microprocessors, with a computational power of around 30 MIPS at 33 MHz. Such microprocessors only take around 50 µs to execute a DES and 300 ms for RSA encryption with a 2048-bit key.

Besides the processor, the different types of memory are the main elements of the microcontroller. They serve to save programs and data. Smartcard microcontrollers are computers in their own right. They generally have between 1 and 8 Kb of RAM, between 64 and 256 Kb of ROM and between 16 and 128 Kb of EEPROM.

The amount of EEPROM available on a smartcard was, for a long time, limited by the fact that EEPROM was not designed specifically for smartcards, and that its physical limits of miniaturization had been achieved. Flash memory has enabled us to overcome this constraint. Thus, we have seen the emergence of the first smartcard prototypes with 1 Mb of persistent Flash memory.

The protection of the smartcard is mainly taken care of by the operating system. The mode of physical routing for data access is only available after personalization of the card just before it is issued to the user. Data are accessed through a logical structure of files secured by access control mechanisms.

The smartcard is widely used in mobile telephone networks (SIM cards), as are public key infrastructures (PKIs). This technology has enabled operators to exploit their network whilst greatly limiting the number of instances of fraud, thereby also ensuring financial viability. It is also the legal support for electronic signatures, as recognized by numerous countries.

The EAP smartcard directly processes the EAP protocol in the smartcard. The main intended applications are EAP-SIM and EAP-TLS. There are many advantages to an EAP-TLS protocol being executed on a smartcard. Firstly, authentication is independent of any given software publisher (e.g. Microsoft). In addition, the security provided is certainly better than with EAP-TLS carried out in software form by the processor of a personal computer, because it is always possible for spyware that has infected the PC to capture the keys. The advantage of smartcards is that all the computations are carried out in the card itself, and the smartcard only outputs an encrypted flow. The secret keys never leave the smartcard.

Schematically, an EAP card provides the following four services:

– *multiple-identity management:* the card holder can use several wireless networks. Each of those networks requires an authentication triplet: EAP-ID (value delivered in the message EAP-RESPONSE.IDENTITY), EAP-Type (type of authentication protocol supported by the network) and cryptographic credits – i.e. the set of keys or parameters used by a particular protocol (EAP-SIM, EAP-TLS, MS-CHAP-V2, etc.). Each triplet is identified by a name (the identity), which can have multiple interpretations (SSID, account username, mnemonic, etc.);

– *assignment of an identity to the card:* the card's identity is contingent upon the host network. Internally, the card may possess several identities, and adapt to the network to which the PC and the smartcard are connected;

– *processing of EAP messages:* as the smartcard possesses a processor and memory, it can execute code and process the EAP messages received and send such messages in response;

– *calculation of the unicast key:* once the authentication session has been completed, the EAP tunnel can be used for the transmission of diverse types of information, such as keys or profiles. It is possible to transmit a session key, for example, and make it available to the terminal wishing to access the resources of the wireless network.

Figure 7.3 illustrates an authentication procedure between an authentication server and an EAP smartcard. The flow of commands passes through the software programs on the PC – i.e. first the EAP software entity, which simply transmits the EAP packets to the RADIUS server, on the one hand, and to the smartcard, on the other – followed by the machine's operating system, which handles the interface with the smartcard, and finally the IEEE 802.11 interface of the wireless link.

To improve security, it is possible to insert chip cards on the server end as well, so that the EAP-TLS algorithm from the authentication server is also run on the chip card. With new chip cards that can store up to 1 Gb, it is possible to memorize the logs needed for traceability.

Clearly, the more clients there are, the more the number of smartcards needs to be increased.

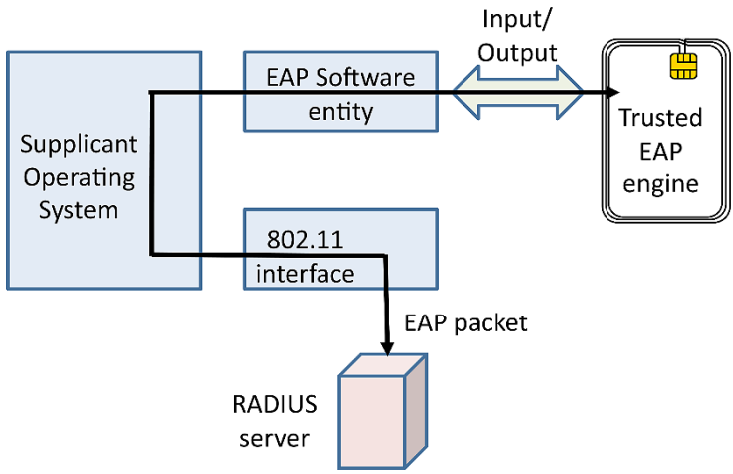



**Figure 7.3.** *Authentication procedure using an EAP smartcard*

## 7.2. Virtual secure elements

Security elements can, themselves, be virtualized by the normal virtualization process: taking a physical machine and installing a hypervisor capable of supporting several virtual smartcards. This process is illustrated by Figure 7.4, with three virtual machines.

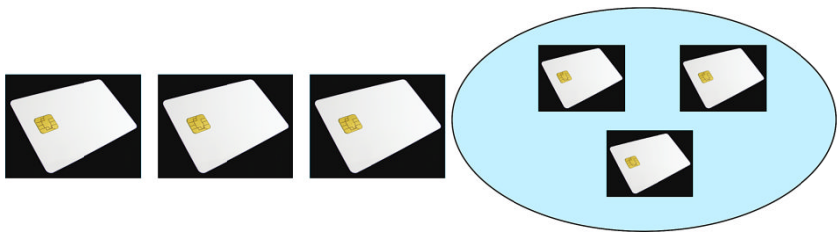


**Figure 7.4.** *Virtualization of smartcards*

However, as we have just described, a hardware element is essential in areas of very high security:

– in software form, keys can always be stolen. It may be difficult to hack certain systems, but there is always a risk, which nobody is willing to take in very high security situations;

– in a secure element, the keys cannot be stolen, or at least not from the card itself. The danger lies in the possibility of hacking whilst the keys are being transported to the secure element.

The solution of virtualization can be viewed completely differently – e.g. a set of physical cards whose functions are deported to a Cloud of security. These secure elements are grouped together, as illustrated in Figure 7.5, on cards; these cards may number in the thousands if necessary.

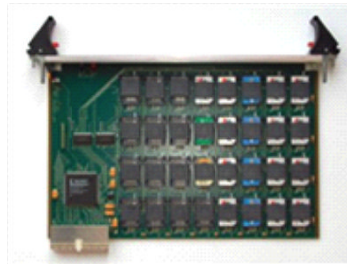


**Figure 7.5.** *A Cloud of secure elements*

One of the paradigms of the new generation of security is to install a secure element associated with each element needing to be defended – be it an individual, an object, a virtual machine or anything else. To access the Internet, clients first need to authenticate themselves with their secure element, irrespective of the object that is connecting. Symbolically, Figure 7.6 represents what is needed by each individual – and, by extension, each thing – connecting to the Internet.

**Figure 7.6.** *The key for the Internet*

Obviously, the symbol could be presented in a completely different form. Before going any further into the access securing processes, let us first examine the different solutions used to provide security to an environment.

## 7.3. The TEE (Trusted Execution Environment)

Figure 7.7 shows the three major possible cases with one form of security based on software, one on hardware and an intermediary form of security pertaining to the TEE (Trusted Execution Environment).

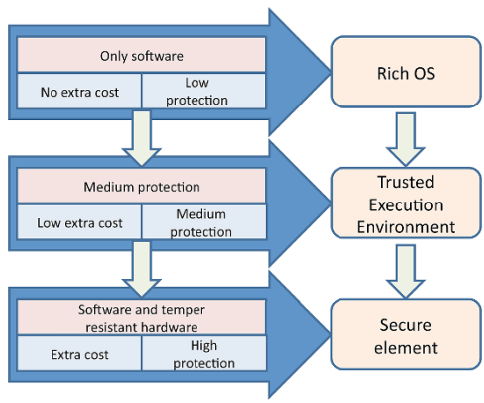



**Figure 7.7.** *The different security solutions*

As we have seen, software security is not costly, but in general it is not very high level. It is often possible for a very good attacker to retrieve a copy and be able to find a certain number of keys. For this reason, it is usual to try and associate a hardware element with the keys. The other end of the security scale is to always have a hardware element to contain keys or important security elements, which can even facilitate the execution of algorithms within the safe box. An intermediary solution, which requires a certain amount of additional explanation, is available on the market today: the TEE (Trusted Execution Environment).

The TEE is a secure zone within the main processor of a smartphone or tablet or any other mobile device, which ensures that sensitive data are stored, processed and protected in a confidential environment. The ability of the TEE to provide safe execution of the authorized security programs, known as “trusted applications”, means it can provide end-to-end security by imposing protection, confidentiality, integrity and access rights on the data.

Smartphone manufacturers and chip manufacturers have developed versions of this technology and built them into their devices as part of their proprietary solution. Thus, application developers need to deal with the complexity of the secure creation and evaluation of the different versions of each application in order to conform to the different sets of specifications and security levels established by each individual proprietary solution.

The first solution to use the TEE is to attach to it a local secure element such as a smartcard, which is found on numerous mobile terminals. The smartcard serves to accommodate the secure part of the applications. The difficulty is in installing several independent applications and being able to modify them, remove them and add new ones with a high level of security. For this purpose, the TSM solution was developed. We shall discuss this solution in the next section.

## 7.4. TSM

A TSM (Trusted Service Manager) is a neutral third party that safeguards the security of the whole procedure of downloading applications (particularly payment accounts) to smartphones with a secure element. Commerce and payment require there to be a certain level of cooperation between mobile operators and financial institutions. The TSM knows the security mechanisms employed by the banks and by mobile telephones, forming the link between multiple financial institutions and operators, whilst guaranteeing complete security of the consumers' credit card information.

The TSM enables a service provider to distribute and remotely manage their contactless applications by allowing access to the secure element via the NFC link. Figure 7.8 shows the relationships between the different actors involved with the TSM.

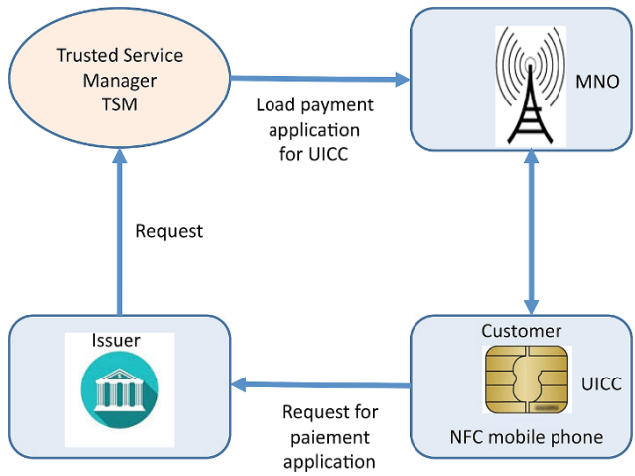



**Figure 7.8.** *The relationships between the different participants with the TSM*

Several TSMs are needed for an issuer – a bank, for example – when it deals with different operators.

Secure elements are indispensable for NFC services to ensure protection of critical applications from the point of view of security,

and to facilitate the use of the same security standards as for debit- and credit cards. However, it is not sufficient to simply integrate a secure element into a mobile telephone. Specific functions are responsible for secure memory allocation so that the zones of the different service providers are separated from one another. Also, it must be possible to establish new secure services (applications and associated cards), on demand, without a third party being able to access the PIN code or other sensitive data. This is the role of the TSM (Trusted Service Manager).

Figure 7.9 offers an overall view of the economic system behind NFC and TSM.

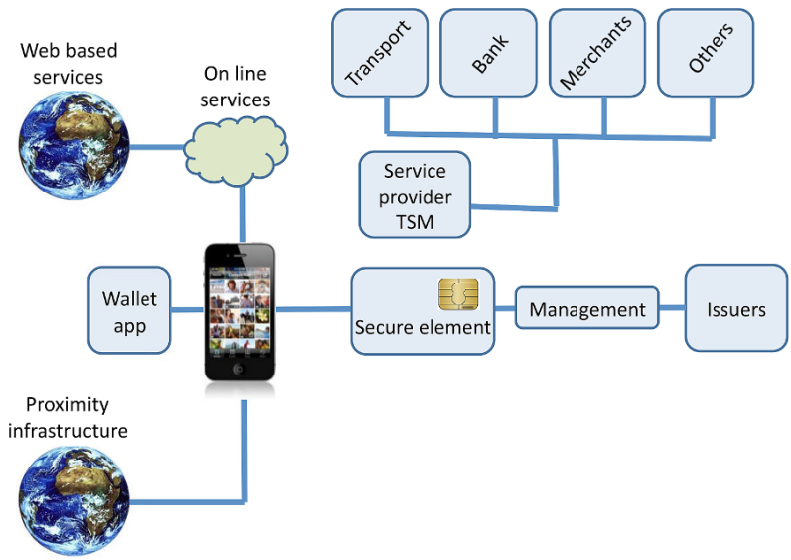



**Figure 7.9.** *The economic system of NFC and TSM*

There are two types of TSM: firstly, SEI TSM (Secure Element Issuer TSM), and secondly, SP TSM (Service Provider TSM). SEI TSM manages the security domains in the whole device, for the service provider and, thus, the whole of the lifecycle of secure elements. SP TSM, for its part, manages the service provider applications available on the secure element. The communication

standards and data transfer to secure domains (SD) are defined by the industrial federation "GlobalPlatform". All types of secure elements are handled. TSMs thus function as sorts of gateways between the different service providers and the end clients.

The TSM infrastructure can also be used to make available and manage critical applications for security which are not based on NFC. Online authentication, as it is used in the context of online and mobile banking services, is an example of this.

One of the challenges that needs to be overcome, in the context of the establishment of the NFC economic system, relates to the fact that the number of secure element providers, and therefore SEI TSM providers, is relatively limited, whereas the number of service providers is growing incessantly. If each service provider had its own TSM, the number of integrations and professional agreements required would simply explode. This is why there are now open SP TSMs which handle several service providers, interacting with most service provider TSMs on the market in question. Service providers have access to a broad range of secure elements, and can thus target most clients using smartphones, whilst secure element manufacturers devote less time and money to the integration of projects including SP TSMs.

As the name indicates, both types of TSM must be trustworthy and, therefore, must satisfy the strictest security standards. This also underlies the implementation of quality processes, such as security measures integrated into the systems and the solutions – in particular, the storage of secure keys. This is critical for the applications involved in payment operations. Payment systems such as Visa or MasterCard thus require a rigorous TSM certification process. Because of these requirements in terms of security and *savoir-faire* in mobile technologies and payment technologies, TSMs, and particularly open SP TSMs, are usually run by a trusted intermediary.

The general problems of the TSM system are as follows:

– neutrality: the TSM must be independent of the issuer and the operator;

– scalability: the TSM must be able to handle all types of credentials for all applications (banks, badges, coupons, transport, etc.);

– the TSM must be able to support all types of form factor such as SIM cards, “secure microcontrollers”, SD cards, eSEs, etc.

The specific problems of the TSM are that:

– the credentials must be isolated for each application from the same issuer and from each different issuer. A variety of options have been defined to create security domains (SD);

– access to the SD is complex. In addition, there may be several secure elements, and it is complex to ensure access to the right secure element;

– multi-tasking is another fairly complex characteristic. No one application should be allowed to obstruct another for too long, on all the security elements. In particular, we need to manage priorities between different issuers;

– the use of SMS or BIP (TCP/IP) for the transport of the applets between the TSM and secure element is problematic. It leads to several problems, including OTA (Over The Air) security, and the creation of SDs in a highly-secure manner;

– the process of installation of the applets must be common to all operators and all issuers;

– there is a DAP (Data Authentication Pattern) mechanism, which enables the issuer to be sure that the installed applet has not been modified by the operator;

– the problem of who is authorized to manage the SDs needs to be precisely defined;

– the personalization of the applets in the SDs needs to be done with caution;

– secure elements have a stringent limitation on the number of SDs. This is a major constraint for the number of NFC applications.

Figure 7.10 shows the security domains of the telecom operator, that of the TSM and those of the applications. All of this is complex to manage, and the rights to the SDs of the operator and of the TSM are imperfectly defined.

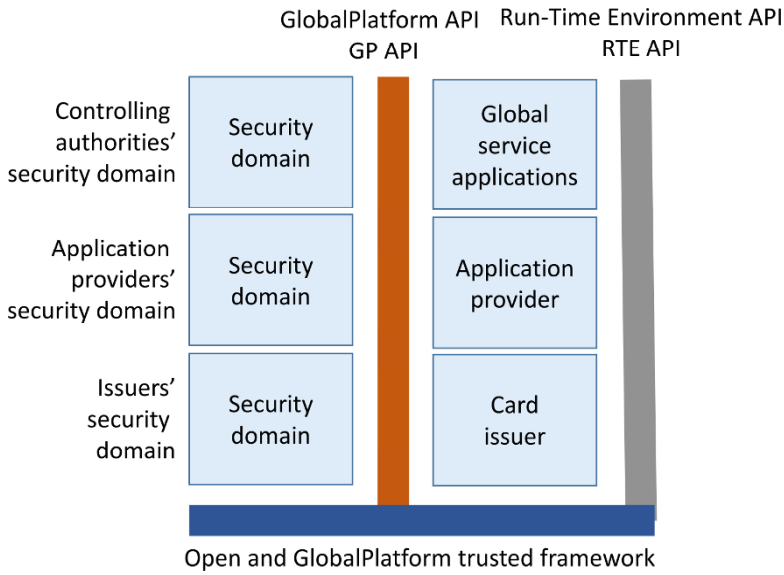



**Figure 7.10.** *The security domains*

## 7.5. Solution without a TSM

A solution without a TSM has been developed by some companies, including Google and EtherTrust. This solution entails deporting the secure elements to servers which can be controlled by the service providers (banks, companies, etc.) or by a security grid provider.

With this solution, all the shortcomings of the current implementation of TSM are eliminated. However, a new constraint is introduced, replacing the drawbacks of TSM: the mobile device must constantly be connected. This solution is represented in Figure 7.11.

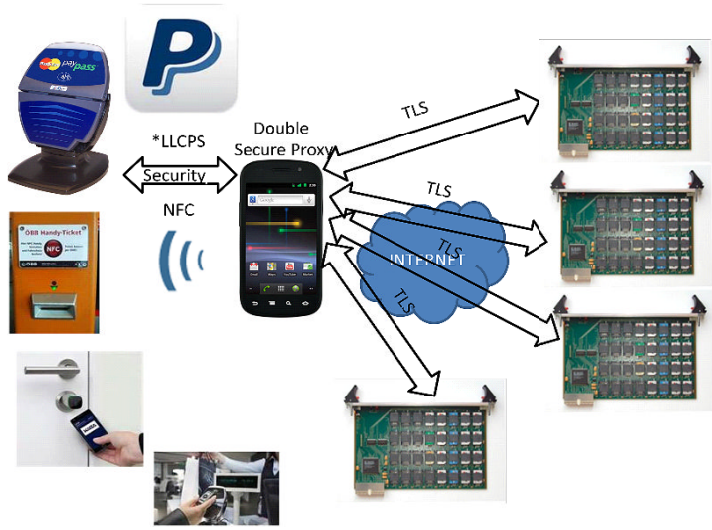



**Figure 7.11.** *Solution without a TSM*

The connection is no longer made with the secure element on the smartphone, but rather with secure elements that are in the card grids. There are as many secure elements as there are applications which a user wishes to achieve. The advantage is that the applets or modifications of the programs are made directly in the secure elements, which are either in NFC service provider or in a secure element provider. This system is scalable, because a user can have as many secure elements and therefore SDs as NFC applications. All the difficulties due to the sharing of the secure element are no longer there. This solution is being extended with the emergence of HCE, which we shall discuss in the next section.

## 7.6. HCE

HCE (Host Card Emulation) is an exact virtual representation of a smartcard using only software. Before the HCE architecture, NFC transactions were mainly made by using a local secure element such as a smartcard. HCE enables us to offer virtual payment cards. These

payment cards can be distributed in real time without having to change the software on the smartphone or tablet.

HCE technology facilitates information transfer across an NFC interface between the NFC component and a remote secure element which can be treated as local. HCE requires the NFC protocol to channel the data to smartphone's operating system instead of to a secure element such as a local smartcard based on hardware configured specifically to respond only as a card, with no other functions.

Android 4.4 and later from Google uses HCE which allows an NFC interface to communicate with the terminal's operating system. Google introduced this platform to carry out secure NFC transactions for payments, loyalty programs, access to external cards and other personalized services. With HCE, any application on an Android 4.4 and later device can emulate an NFC smartcard, enabling users to initiate transactions with an application of their choice. The applications can also use a new type of reader, to act as a reader of HCE cards and other NFC devices.

## 7.7. Securing solutions

Having described TEE, TSM and HCE, we can now go on to describe the different effective solutions in terms of access security using secure elements. There are two opposing solutions: the local solution, with a secure element in the terminal or connected directly to it, and the delocalized solution, which is based on virtualized secure elements.

Figure 7.12 illustrates the first solution, whereby a secure element must be available locally. That secure element could be a SIM card from a telecom operator, the inbuilt secure element from the manufacturer that made the terminal, a secure element hosted on an SD card and inserted into the reader of the mobile terminal, or even an external secure element, communicating with the mobile via an NFC interface, for example. These different solutions developed

greatly between 2010 and 2015, because they represented practically the only solution, before the arrival of HCE. Manufacturers who have been able to make enough of a name for themselves in the market of smartphones, tablets and portable electronics have, of course, chosen to employ this solution. The secure element is a smartcard, often made by Gemalto, or an inbuilt secure element – made by NXP for example – as is the case in Apple's latest smartphones.

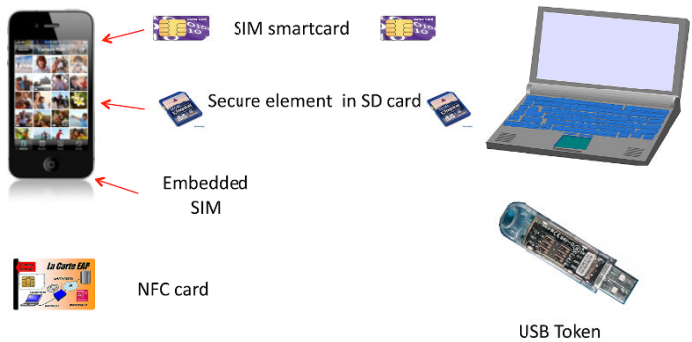



**Figure 7.12.** *Securing by local secure elements*

This solution is easy to implement, with the advantage of being able to carry out communications locally, but the disadvantages listed in the section on TSM – namely the significant limitation of the number of programs that can be embedded in a secure element and the difficulty of configuring the secure element when a new service is added.

The second major solution is the "remote smartcard", which is advantageous if we want high security, effectively using a secure element. This solution has become possible because of the arrival of Android's HCE. Before, the solution used was always the first type, as indicated in Figure 7.13.

With the advent of HCE, the NFC component can be directly connected to an external secure element, as indicated in Figure 7.14.

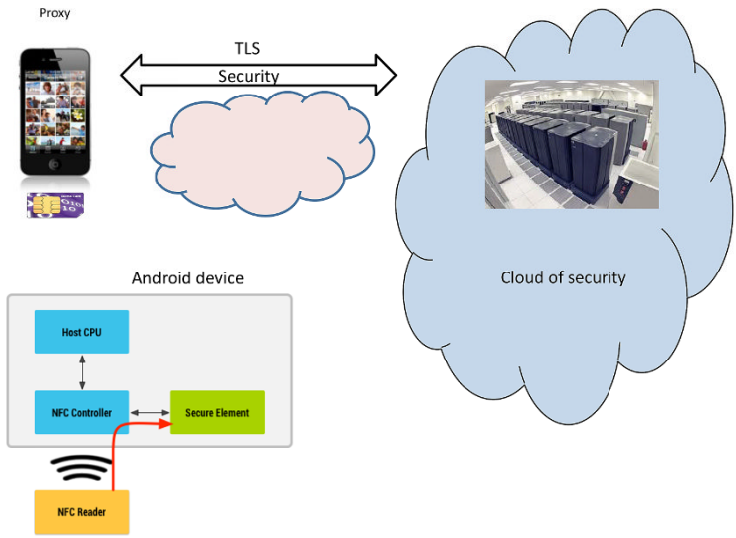



**Figure 7.13.** *Securing using external secure elements before the Android 4.4*

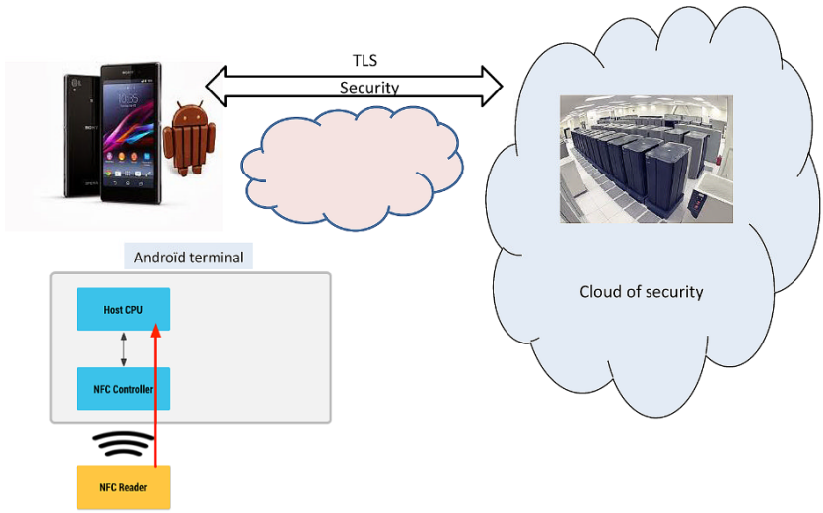



**Figure 7.14.** *Securing using external secure elements with Android 4.4 and later*

Before giving a more detailed description of this solution and presenting examples of its operation with classic applications, let us

look at how confidential communication takes place between the mobile terminal and, say, an Internet-based trader. This relationship is illustrated in Figure 7.15.

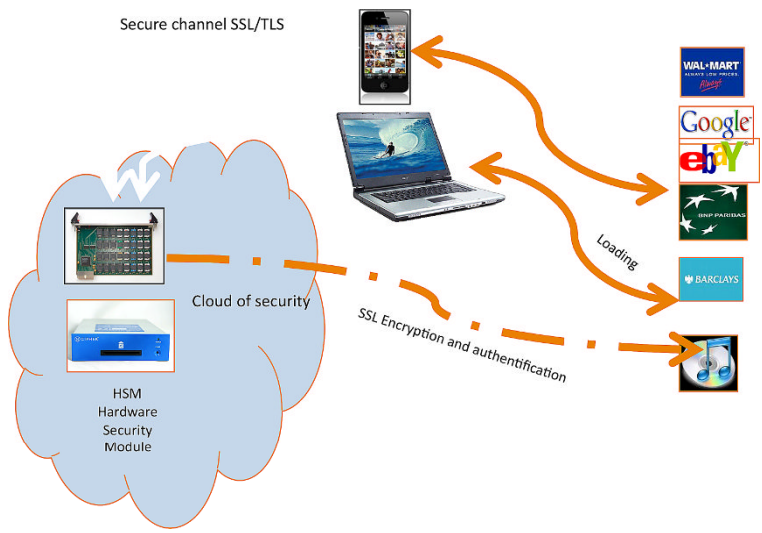



**Figure 7.15.** *Securing by a Cloud of secure elements*

Communication takes place between the mobile terminal and the secure element, which is in the Cloud of security. This communication takes place after the opening of a secure tunnel once contact has been established with the SSL/TLS protocol. Once this communication has been completed, the secure element always connects with the trader through a secure channel which is opened between the secure element and the retailer's site. Purchases can then be made perfectly securely. Once the purchase has been made, control is passed back to the mobile terminal, which is able to retrieve the bought music or video or, indirectly, any type of purchase. The secure element may be in a Cloud of secure elements, but also in an HSM (Hardware Security Module) which plays the same role, but the client still needs to trust that device. It is for this reason that Clouds of secure elements are still the most likely solution to develop, because secure elements – e.g. smartcards – may actually be held by the user, who will therefore have a high level of trust in that system.

One initial advantage of this solution using a virtualized or decentralized smartcard is the possibility of assigning to a particular user not just one smartcard but however many s/he actually needs. The client can have multiple banks, multiple key managers, and multiple security service providers. A security service provider can quite easily carve out a niche in this market by setting up its own secure-element server or hosting secure elements with a Cloud-of-security provider. This property is illustrated in Figure 7.16.

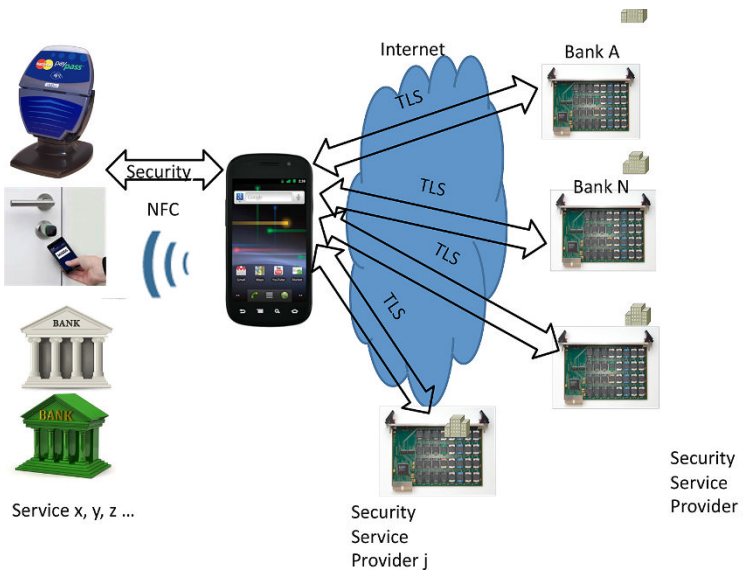



**Figure 7.16.** *Advantages of the external solution*

To be complete, the environment must have its own secure-element management system. For this purpose, a simple solution is offered by the Global Platform consortium, involving adding an administration center which can, itself, be perfectly well virtualized. Figure 7.17 shows the environment, complete with its administration server.

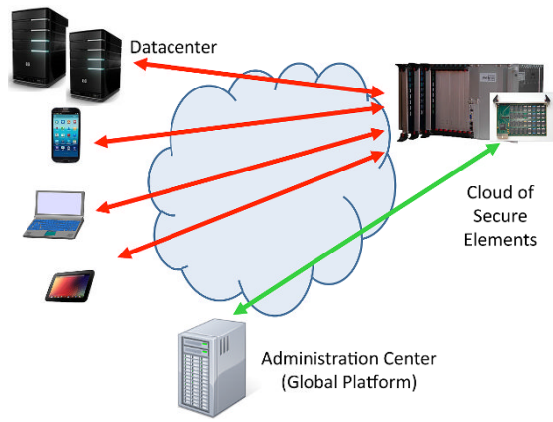



**Figure 7.17.** *Architecture of securing using external secure elements*

Numerous security services can be constructed using the above model, drawing on a Cloud of security. We have already seen the examples of virtual firewalls and virtual DPI. With regard to the Cloud of secure elements, we can cite management of the identity of virtual machines, and management of hotel, car and house keys, and more generally, any keys we may need to protect something. Figure 7.18 represents the securing of virtual machines. Indeed, today, we are seeing increasing numbers of attacks launched from Clouds, such as the creation of specific virtual machines for attacking private servers, which may be external or internal. In the internal case, we can cite attacks which, in an attacking virtual machine, randomly forge all the frames corresponding to a protocol based, for instance, on the RFC describing that protocol in detail. These random frames are then sent to the datacenter's internal bus, with a very high probability that one of those frames will be an attack frame and will take down one or more servers which, as they shut down, will force other servers to do so too.

Another application is that of mobile keys which are stored in a Cloud of security, which can open a door when the smartphone is held next to the lock. The communication is established directly between the Cloud of security and the NFC lock. Of course, we must be aware of the possibility of an occasional communication breakdown – for

example, when we cannot connect to the Cloud of security at the moment of presenting the smartphone to the lock. In such a case, the key can be stored in the TEE of the mobile devices for a limited period of time.

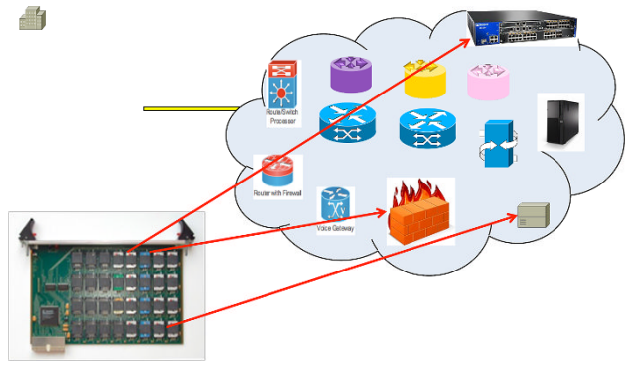



**Figure 7.18.** *Securing of virtual machines*

The applications of purchasing and the relationship with a banking site are also part of the services that can be provided by this virtual chip card solution. One of the types of transactions is illustrated in Figure 7.19. The transaction takes place between the NFC reader of the retailer site and the user's secure element which – in this figure – is in a smartcard in a Cloud of security. As in the case of the mobile key, the mobile phone merely serves as a modem to facilitate the transaction between the two endpoints and establish the necessary secure channels. This solution is sufficient for limited-cost purchases. For larger transactions, it is necessary to add further communications with the "issuer" and the bank itself.

In conclusion to this chapter, we have seen a new approach to securing by the use of secure elements. This approach is one of the key security solutions in the world of telecommunications and the Internet. Having introduced secure-element-based techniques, we described the potential for innovations offered by the Cloud of security.

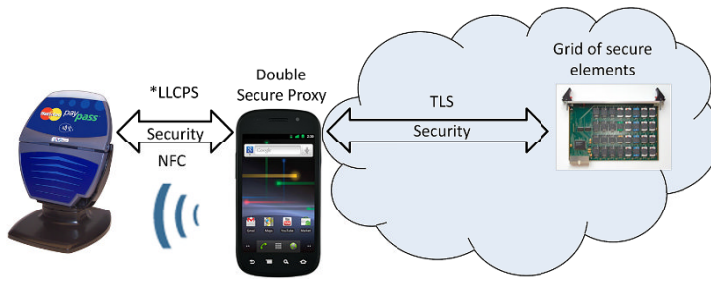



**Figure 7.19.** *Securing of an electronic payment*

Service providers and operators have a choice between the two approaches: local, with a secure element in the mobile terminal, or remote, with a secure element in the Cloud of security. Usually, the choice is dictated by whether or not it is possible to have a secure element built into the mobile terminal. In 2015, Apple opted to use local secure elements in light of their extensive range of smartphones. Apple has been able to launch applications based on this approach, such as Apple Pay. Google, on the other hand, opted for the external secure element solution after they failed to obtain a sufficient range of smartphones by buying Motorola. (This is why they then re-sold Motorola). Nevertheless, Google is looking into both solutions – local and delocalized – having re-entered the mobile phone market.

## 7.8. Conclusion

The world of security is very extensive, and it has only been possible, in this chapter, to offer a very partial view of network security. In particular, we have focused on the new generation, linked to the Cloud, with the Cloud of security. This solution has advantages and disadvantages, but it is on new ground: that of easy entry into the market and easy tailoring to the security objective. Only time will tell how much of a market share it will win.

# 8

# Concretization and Morphware Networks

Concretization is the reverse of virtualization: the question becomes one of how to make the move from software to hardware but preserve the same degree of agility as software. In other words, the idea of concretization is to replace the software with hardware that is instantaneously reconfigurable, so that the software running on the reconfigurable processor can instantly be replaced by different software. Figure 8.1 illustrates the relationship between the processes of concretization and virtualization. However, it should be understood that the hardware which is the startpoint of virtualization is nothing like the hardware produced by concretization. For example, the original hardware might be a router, and that router cannot be replaced by another machine. Concretization produces hardware which can be modified instantaneously (or almost instantly) so as to become a different device from the current one. Obviously, the objective is to speed up the execution of the software obtained by virtualization.

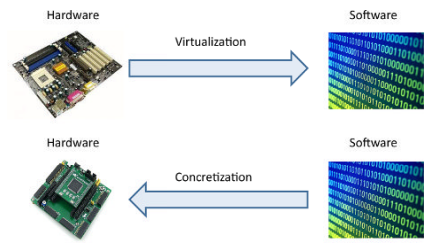



**Figure 8.1.** *The process of concretization*

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

## 8.1. Accelerators

One initial solution to speed up the computation of software is represented by accelerators. Many different types of performance accelerators are already on the market, including DPDK (Data Plane Development Kit). The DPDK environment is a collection of data plane libraries and network interface drivers which accelerate communications. Real migrations of virtual machines – i.e. the actual transport of all the software associated with a virtual machine, from one physical machine to another – can be done in the space of a few milliseconds.

The purpose of DPDK, as we have seen, is to create a set of libraries for environments made up of software which are designed to perform specific functions, requiring large amounts of computation power. This type of software is found in the context of virtualization for signal-processing applications or multimedia protocol management. In particular, this solution applies in the networking domain by way of an additional: the EAL (Environment Abstraction Layer). The EAL masks the peculiarities of the environment, by presenting a programming interface to libraries, interfaces, hardware accelerators and operating-system elements such as Linux or FreeBSD. Once the EAL has been created for a specific environment, developers string together the libraries and interfaces to create their own applications. For example, the EAL can provide a standard framework to handle Linux, FreeBSD, Intel IA 32- or 64-bit, or IBM Power8.

The EAL also provides additional services, such as temporal references, access to the PCIe bus, tracking functions, debugging solutions and alarm functions. Finally, the DPDK environment implements a model with very little overhead, which helps to obtain excellent performance in terms of the data plane. The environment also enables us to access the different nodes by eliminating the computation overheads, leading to rather impressive accelerations which are achieved by way of fairly simple processes.

The DPDK environment also includes examples which demonstrate best practices for software architectures, tips for the

design of data structures and storage, tuning of applications and advice on how to overcome the performance deficits of different solutions for network deployment.

DPDK is one of the best examples of accelerators to compensate for the deficient performance of processors running software which needs to achieve performances compatible with its function. There are other types of accelerators, which can be seen as being intermediary between entirely software-based solutions and purely hardware solutions. However, the advent of increasingly-powerful reconfigurable microprocessors certainly represents the future of concretization far better.

## 8.2. A reconfigurable microprocessor

A reconfigurable microprocessor is a microprocessor with erasable hardware that can be rewired dynamically. This enables the chip to adapt effectively to the programming tasks required by a particular piece of software. For example, the reconfigurable processor can be transformed from a video card to a graphics card – both of them being optimized to enable the applications to run at the highest possible speed. We can say that they are bespoke processors. In practical terms, this ability results in a high degree of flexibility in terms of the chip's functions. For example, a single chip could be used simultaneously for a camera, a tape player, a signal processor, a router and a firewall. We need only load the desired software and the processor is reconfigured automatically to optimize the performances necessary for the programmed function.

Several types of reconfigurable processors are available on the market. First of all, there are DSPs (Digital Signal Processors), which exhibit excellent performance. DSPs are programmable chips used in cellphones, automobiles and many music- and video players. Another version of reconfigurable microprocessors has programmable memory matrices which perform hardware functions using software tools. These microprocessors are more flexible than specialized DSPs, but are also slower and more expensive. Hardwired chips are oldest, least expensive and fastest, but unfortunately the least flexible. A DSP is illustrated in Figure 8.2.

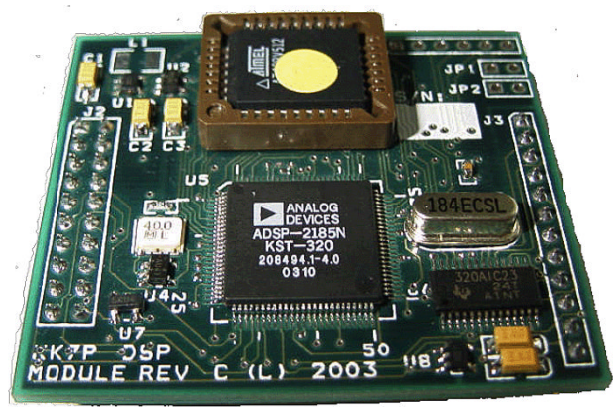



**Figure 8.2.** *A DSP*

Reconfigurable microprocessors have different names depending on exactly what is reconfigured. The best-known examples are FPGAs (Field-Programmable Gate Arrays), which are arrays of gates that can be programmed using RAM technology. Figure 8.3 shows an example of an FPGA.

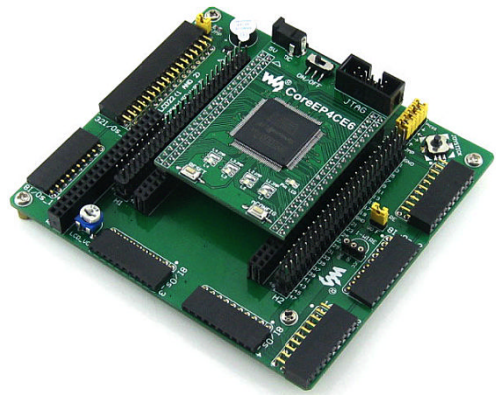


**Figure 8.3.** *An example of an FPGA*

EPLDs (Erasable Programmable Logic Devices) are also programmable logic circuits, but they use FLASH technology.

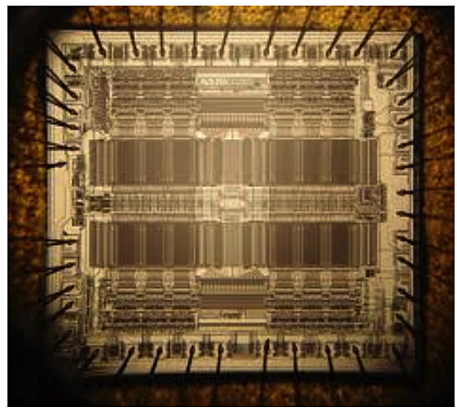


**Figure 8.4.** *An EDLP component*

We can represent the different types of microprocessors in the form shown in Figure 8.5.

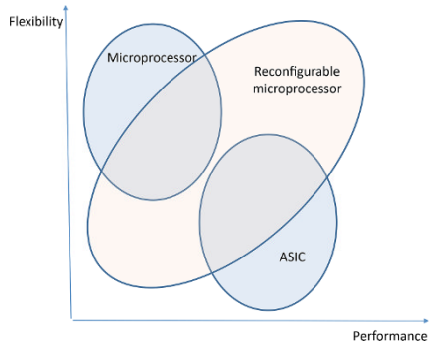



**Figure 8.5.** *The different types of microprocessors*

Reconfigurable microprocessors form a new generation of systems which, in the long term, should come to replace all software elements requiring a little computing power. These reconfigurable microprocessors comprise a layer of hardware elements and a layer of memory elements. The hardware element has a varying degree of complexity, represented by its "granularity". Granularity ranges from fine to coarse, to express the complexity of the hardware element. For a fine grain, we work at bit level, so we can program absolutely any

logical function. However, this solution is costly in terms of performance, and is not able to deliver all the desired orders of magnitude of performance, for certain computational components in networks. In addition, the reconfiguration time is too long to handle several real-time processes on the same reconfigurable processor.

In the case of coarse grains, it is no longer possible to perform all functions directly. The elements form operators, which can directly be used for the necessary operations in signal processing or multimedia protocols. These operators can be reconfigured much more quickly, because they are limited in terms of the number of functions that the component itself can perform.

More specifically, the granularity of the reconfigurable element is defined as the size of the smallest basic block – the CLB (Configurable Logic Block) – which can be included in the string of functions to be performed. A high degree of granularity – i.e. fine granularity – means great flexibility to implement the algorithms using hardware. We can implement almost any type of function. We use fine grains to carry out particular functions or test new algorithms, before moving on to coarser grains. The difficulties facing fine-grained circuits include the higher power requirement and the slower execution speed, due to the path to be followed, which is generally quite long. Reconfiguration may also require a lot of time in comparison with the time-periods necessary to maintain a real-time process. On the other hand, coarse grains have much shorter chain paths and lend themselves more easily to real-time applications.

Of course, it is important that the computation of a function correspond as closely as possible to the path followed. If the granularity is too coarse, there is a risk that the component will take more time than is necessary – in other words, poor use coupled with higher consumption. For example, an addition on four bits performed on a component with a granularity of sixteen bits degrades performance, as significantly more resources are consumed.

The idea to find the best compromise is to make matrices of coarse-grained elements mixed with fine-grained elements. We can

obtain such an arrangement on a single chip with rDPAs (reconfigurable Datapath Arrays) and FPGAs.

Matrix architectures combining rDPA and FPGA must be optimized so that the path taken between the different elements is as short as possible. If the architecture is not appropriate, we immediately see a significant loss of efficiency, but this is not necessarily a major problem, if the overall circuit serves numerous, very different algorithms. In fact, the best scenario is to know, in advance, which algorithms will be executed on the circuit, with a view to being able to optimize the logic elements, the memory and the path that is followed. FPGAs are generally too fine-grained to be highly efficient in terms of performance, and they need to be substituted with coarser-grained elements that are better suited to the purposes of the reconfigurable microprocessor.

Reconfiguration can take place when the circuit is engaged, between two phases of execution, or even during the execution of a process. Circuits which work at the bit level, such as FPGAs, require a lot of time for reconfiguration, which is carried out by way of a bitstream that is fairly complex in relation to coarse-grained architectures, which require a much shorter binary stream, with much quicker establishment. In fact, most reprogrammable cards have partial reconfigurations that facilitate the execution of a function whilst another part of the microprocessor continues to execute a function.

Reconfigurable element matrices seem to represent the best compromise between fine- and coarse-grained architectures, bringing together enough circuits to find optimal chain paths. To control this collection of circuits, a host processor is needed. The host needs to be powerful to determine which algorithms to use to compute the chain. Overall, the flexibility of the architecture hinges on the path taken to interconnect the gates. Thus, the path to be determined is of prime importance in carrying out complex functions, such as those found in certain protocol- or signal processing operations.

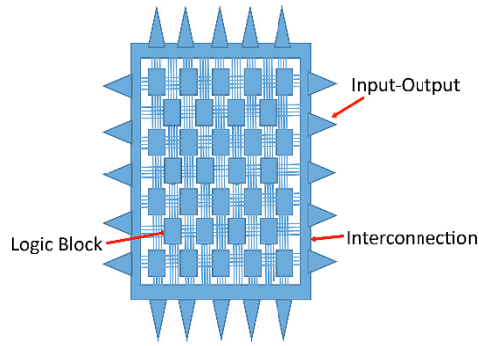



**Figure 8.6.** *Reconfigurable element matrix*

In actual fact, we need to be able to have fine-grained and coarse-grained reconfigurable components in a single system, and to combine them in a programmable environment such as that illustrated in Figure 8.7.

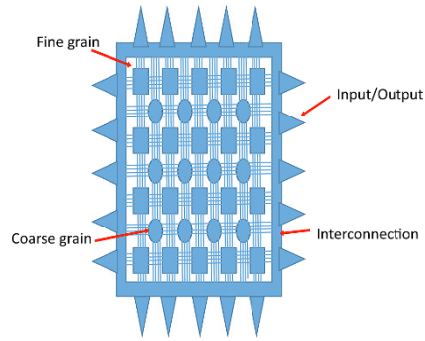



**Figure 8.7.** *Reconfigurable microprocessor using fine- and coarse-grained elements*

The datacenter environments of the future should be built around reconfigurable memories and microprocessors, as we can see.

## 8.3. Morphware networks

Looking at the history of networks, the story began with hardware networks – i.e. networks whose nodes are formed of hardware, as

illustrated by Figure 8.8. The nodes are not in the least bit agile; they are as they are, and if we wish to increase their power, we need to upgrade the hardware, and if we want to change technology – e.g. to go from using routers to using switches – again we need to change the hardware. Thus, there is no flexibility in this solution, and yet it is this solution which has been employed practically since the dawn of networking.

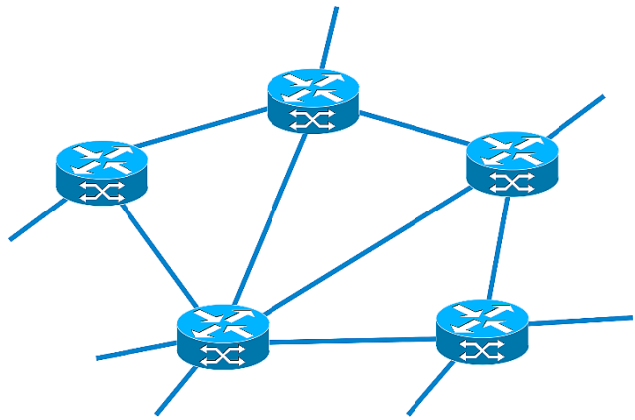


**Figure 8.8.** *A hardware network*

The virtualization of network equipment has facilitated the introduction of software networks. These networks, as we have seen throughout this book, offer a very high degree of flexibility. The size of the devices can be adapted to the traffic on the network, and the technologies used can very easily be changed. The same physical node is used to accommodate multiple virtual nodes, known as software nodes. The number of software nodes depends on numerous parameters, such as the power of the node's processor, the available memory space, the input/output ports, etc. A set of software networks can be put in place to serve specific applications. This category of networks is illustrated in Figure 8.9.

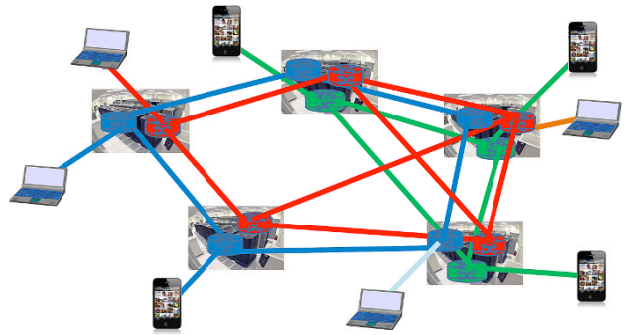


**Figure 8.9.** *Software networks. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

In software networks, the networking machines that are virtualized can move, to go to a physical node that has more appropriate capacity, or following the breakdown or malfunction of the original node. The advantage of such software networks is the agility of the nodes and the great simplicity of installation and change. There are also numerous disadvantages, such as the management of each software network, the sizing of the software networks, and security.

Yet an important point which is rarely discussed relates to the performance of software networks. The fact of moving from hardware to software decreases performance by several orders of magnitude. Of course, the power of the datacenters is such that it is possible to make up for a loss of 10 to 100 orders of magnitude quite easily. So, it is possible to make up entirely for the loss in performance. But it is more difficult to compensate for a greater loss than this, because we need to find software accelerators which are not always easy to manage. For this reason, we are witnessing the rise of so-called "morphware networks" – i.e. networks which change depending on the client and adapt their performance to what is necessary in order for the network to operate in the best possible conditions. In fact, each client is assigned an individual network, and the network's performance is adapted to the client. The idea is to use concretization to obtain the requisite performance. The physical nodes of the network are datacenters of varying dimensions, but these datacenters contain microprocessors of the three different types encountered at the start of this chapter: processors as found in datacenters today, capable of

handling bespoke software. We also find ASICs, capable of handling very particular real-time processes, such as signal-processing. Finally, there are also a fairly large number of reconfigurable processors, so that the necessary reconfigurations do not interfere with the processes occurring in the node.

These new morphware networks are illustrated in Figure 8.10. They represent a new generation, primarily based on reconfigurable processors. This position means they can be used as intermediaries between physical networks and software networks.

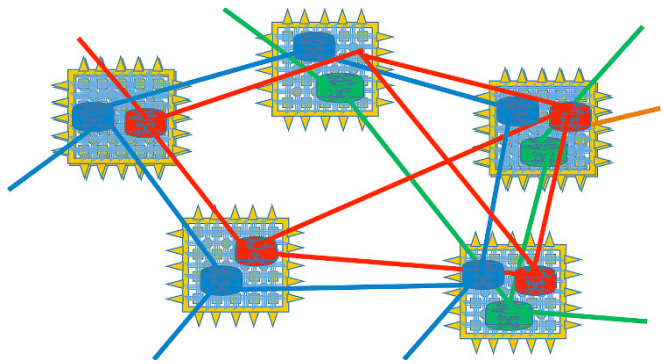


**Figure 8.10.** *Morphware network. For a color version of the figure, see www.iste.co.uk/pujolle/software.zip*

In conclusion to this chapter, we have examined the question of how to improve the performance of software networks, whose power is lower than that of hardware networks. With a view to preserving the flexibility and agility of software networks but with the power of hardware networks, concretization is becoming a hugely important process. By concretization, we are able to obtain networks which are perfectly suited to the services and applications needing to be served.

## 8.4. Conclusion

This chapter looks at the likely developments in the longer term, with concretization and networks that mutate to suit the clients' needs. There are numerous problems that need to be solved, and new ideas

have been put forward, including the creation of a software network every time a connection request is made. Whilst such networks cannot be implemented on a true scale today, there is no reason to think that the same will be true in a few years' time. Hence, rapid reconfigurable processors would represent an excellent solution.

# Conclusion

The networks of the future will be completely virtualized in physical infrastructures – essentially datacenters of varying sizes. Very small datacenters will be located on the periphery of the network, near to the user. This physical infrastructure will be used to support software networks that are tailored to the clients' needs and to those of the applications for which they were generated. The agility of these networks is the main differencewith previous-generation networks: it is possible to replace a network in a few minutes, or even a few seconds, and by using automation, in a few milliseconds.

However, it is important to note the difficulties that could arise from this new generation of networks: the complexity of managing the different networks and the security of the whole system. Indeed, for reasons of isolation, management cannot be shared, for fear that one network will be intermingled with another. Security is also a major issue, because of the increased complexity of the architecture and the diversity of the networks. The Cloud of security is a promising new paradigm, but it does not solve all the problems.

Overall, this new generation of technologies is based on the Cloud, and on virtualization, as is indicated by Figure C.1, which also shows the overlap with migration, NFV and Cloud of security.

*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

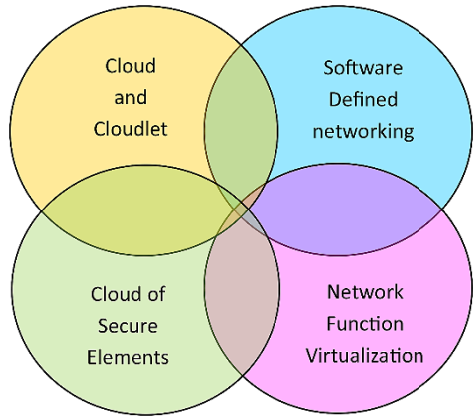



**Figure C.1.** *The fundamental elements of new generation networks (NGNs)*

The NFV standard attaches a great deal of hope to the major simplification which is offered by virtual machines. The Open Platform for NFV (OPNFV)project represents another promising avenue. The OPNFV solution would be Carrier-Grade, integrated, and should give rise to an open-source platform developed by the Linux Foundation.

Another significant advance would be concretization – i.e. the opposite of virtualization. Indeed, with software networks, we tend more to see a drop in performance in comparison to physical networks, even though the power of Clouds limits that reduction. Companies such as Radisys and Intel are beginning to explore the option of hardware that behaves like software.

5G is also greatly affected by virtualization, which offers numerous functions that are indispensable for a simple connection of billions of things. Virtualization is even responsible for massive revolutions, such as the use of C-RAN techniques, whichis spreading across the globe. Amongst these revolutions, a local loop could become very simple indeed, with a potentialreturn to the use of analog signals, which would be transported directly to the Cloud and computed there. Certainly, it is true that far more information can be conveyed by an analog signal than a digital one.

Finally, there is a new paradigm in the area of security, with a certain virtualization of secure elements, but this is more a question of delocalization of the secure element than of actual virtualization in the truest sense of the word.

# Bibliography


[AVR 14] AVRAMOV L., PORTOLANI M., *The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology*, CISCO Press, 2014

[AZO 13] AZODOLMOLKY S., *Software Defined Networking with OpenFlow*, Packt Publishing, 2013

[GEN 14] GENG H., *Data Center Handbook*, Wiley, 2014

[GOR 14] GORANSSON P., BLACK C., *Software Defined Networks: A Comprehensive Approach*, Morgan Kaufmann, 2014

[HOO 14] HOODA S.K., KAPADIA S., *Using TRILL, FabricPath, and VXLAN: Designing Massively Scalable Data Centers (MSDC) with Overlays*, CISCO Press, 2014

[HU 14] HU F., *Network Innovation through OpenFlow and SDN: Principles and Design*, CRC Press, 2014

[LOW 13] LOWE S., MARSHALL N., GUTHRIE F. *et al.*, *Mastering VMware vSphere 5.5*, SYBEX, 2013

[MAD 14] MADISETTI V., BAHGA A., *Internet of Things (A Hands-on-Approach)*, VPT publisher, 2014

[MIR 14] MIR N. F., *Computer and Communication Networks*, Prentice Hall, 2014

[MOR 14] MORREALE P.A., ANDERSON J.M., *Software Defined Networking: Design and Deployment*, CRC Press, 2014

[NAD 13] NADEAU T.D., GRAY K., *SDN: Software Defined Networks*, O'Reilly, 2013



*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

[RHO 14] RHOTON J., DE CLERCQ J., NOVAK F., *OpenStack Cloud Computing: Architecture Guide*, RP publisher, 2014

[SAB 13] SABOOWALA H., ABID M., MODALI S., *Designing Networks and Services for the Cloud: Delivering business-grade cloud applications and services*, CISCO Press, 2013

[SHU 13] SHUKLA V., *Introduction to Software Defined Networking - OpenFlow & VxLAN*, Createspace, 2013

[WEN 14] WEN H., TIWARY P.K., LE-NGOC T., *Wireless Virtualization*, Springer, 2013

[YEL 14] YELURI R., CASTRO-LEON E., *Building the Infrastructure for Cloud Security: A Solutions View*, Apress Open, 2014

# Index



*Software Networks*: *Virtualization, SDN, 5G and Security*, First Edition. Guy Pujolle.

## N, S, T

Other titles from

in

Networks and Telecommunications

## 2015

BENSLAMA Malek, KIAMOUCHE Wassila, BATATIA Hadj
*Connections Management Strategies in Satellite Cellular Networks*

BENSLAMA Malek, BATATIA Hadj, BOUCENNA Mohamed Lamine
*Ad Hoc Networks Telecommunications and Game Theory*

## 2014

ANJUM Bushra, PERROS Harry
*Bandwidth Allocation for Video under Quality of Service Constraints*

BATTU Daniel
*New Telecom Networks: Enterprises and Security*

BEN MAHMOUD Mohamed Slim, GUERBER Christophe, LARRIEU Nicolas, PIROVANO Alain, RADZIK José
*Aeronautical Air−Ground Data Link Communications*

BITAM Salim, MELLOUK Abdelhamid
*Bio-inspired Routing Protocols for Vehicular Ad-Hoc Networks*

CAMPISTA Miguel Elias Mitre, RUBINSTEIN Marcelo Gonçalves
*Advanced Routing Protocols for Wireless Networks*

CHETTO Maryline
*Real-time Systems Scheduling 1: Fundamentals*
*Real-time Systems Scheduling 2: Focuses*

EXPOSITO Ernesto, DIOP Codé
*Smart SOA Platforms in Cloud Computing Architectures*

MELLOUK Abdelhamid, CUADRA-SANCHEZ Antonio
*Quality of Experience Engineering for Customer Added Value Services*

OTEAFY Sharief M.A., HASSANEIN Hossam S.
*Dynamic Wireless Sensor Networks*

PEREZ André
*Network Security*

PERRET Etienne
*Radio Frequency Identification and Sensors: From RFID to Chipless RFID*

REMY Jean-Gabriel, LETAMENDIA Charlotte
*LTE Standards*
*LTE Services*

TANWIR Savera, PERROS Harry
*VBR Video Traffic Models*

VAN METER Rodney
*Quantum Networking*

XIONG Kaiqi
*Resource Optimization and Security for Cloud Services*

## 2013

ASSING Dominique, CALÉ Stéphane
*Mobile Access Safety: Beyond BYOD*

BEN MAHMOUD Mohamed Slim, LARRIEU Nicolas, PIROVANO Alain
*Risk Propagation Assessment for Network Security: Application to Airport Communication Network Design*

BEYLOT André-Luc, LABIOD Houda
*Vehicular Networks: Models and Algorithms*

BRITO Gabriel M., VELLOSO Pedro Braconnot, MORAES Igor M.
*Information-Centric Networks: A New Paradigm for the Internet*

BERTIN Emmanuel, CRESPI Noël
*Architecture and Governance for Communication Services*

DEUFF Dominique, COSQUER Mathilde
*User-Centered Agile Method*

DUARTE Otto Carlos, PUJOLLE Guy
*Virtual Networks: Pluralistic Approach for the Next Generation of Internet*

FOWLER Scott A., MELLOUK Abdelhamid, YAMADA Naomi
*LTE-Advanced DRX Mechanism for Power Saving*

JOBERT Sébastien *et al.*
*Synchronous Ethernet and IEEE 1588 in Telecoms: Next Generation Synchronization Networks*

MELLOUK Abdelhamid, HOCEINI Said, TRAN Hai Anh
*Quality-of-Experience for Multimedia: Application to Content Delivery Network Architecture*

NAIT-SIDI-MOH Ahmed, BAKHOUYA Mohamed, GABER Jaafar, WACK Maxime
*Geopositioning and Mobility*

PEREZ André
*Voice over LTE: EPS and IMS Networks*

## 2012

AL AGHA Khaldoun
*Network Coding*

BOUCHET Olivier
*Wireless Optical Communications*

DECREUSEFOND Laurent, MOYAL Pascal
*Stochastic Modeling and Analysis of Telecoms Networks*

DUFOUR Jean-Yves
*Intelligent Video Surveillance Systems*

EXPOSITO Ernesto
*Advanced Transport Protocols: Designing the Next Generation*

JUMIRA Oswald, ZEADALLY Sherali
*Energy Efficiency in Wireless Networks*

KRIEF Francine
*Green Networking*

PEREZ André
*Mobile Networks Architecture*

## 2011

BONALD Thomas, FEUILLET Mathieu
*Network Performance Analysis*

CARBOU Romain, DIAZ Michel, EXPOSITO Ernesto, ROMAN Rodrigo
*Digital Home Networking*

CHABANNE Hervé, URIEN Pascal, SUSINI Jean-Ferdinand
*RFID and the Internet of Things*

GARDUNO David, DIAZ Michel
*Communicating Systems with UML 2: Modeling and Analysis of Network Protocols*

LAHEURTE Jean-Marc
*Compact Antennas for Wireless Communications and Terminals: Theory and Design*

RÉMY Jean-Gabriel, LETAMENDIA Charlotte
*Home Area Networks and IPTV*

PALICOT Jacques
*Radio Engineering: From Software Radio to Cognitive Radio*

PEREZ André
*IP, Ethernet and MPLS Networks: Resource and Fault Management*

TOUTAIN Laurent, MINABURO Ana
*Local Networks and the Internet: From Protocols to Interconnection*

## 2010

CHAOUCHI Hakima
*The Internet of Things*

FRIKHA Mounir
*Ad Hoc Networks: Routing, QoS and Optimization*

KRIEF Francine
*Communicating Embedded Systems / Network Applications*

## 2009

CHAOUCHI Hakima, MAKNAVICIUS Maryline
*Wireless and Mobile Network Security*

VIVIER Emmanuelle
*Radio Resources Management in WiMAX*

## 2008

CHADUC Jean-Marc, POGOREL Gérard
*The Radio Spectrum*

GAÏTI Dominique
*Autonomic Networks*

LABIOD Houda
*Wireless Ad Hoc and Sensor Networks*

LECOY Pierre
*Fiber-optic Communications*

MELLOUK Abdelhamid
*End-to-End Quality of Service Engineering in Next Generation Heterogeneous Networks*

PAGANI Pascal *et al.*
*Ultra-wideband Radio Propagation Channel*

## 2007

BENSLIMANE Abderrahim
*Multimedia Multicast on the Internet*

PUJOLLE Guy
*Management, Control and Evolution of IP Networks*

SANCHEZ Javier, THIOUNE Mamadou
*UMTS*

VIVIER Guillaume
*Reconfigurable Mobile Radio Systems*