Quality of Service Requirements System Satisfaction for new approach of Wireless Communication

Dr. Hamood Shehab Hamid

Middle Technical University /Electrical Engineering Technical College /Department of Computer Engineering Techniques /drhamood@mtu.edu.iq /Baghdad-Iraq

ABSTRACT:

During our autopsy, we studied and identified the main components of local wireless networks and presented some of the work of the scientific community on these same networks. Nevertheless, and despite the ambition of the MANET group, it is clear that all contributions on wireless networks is focused on the management of physical layers, access methods, mobility, routing. This is partly due to the complexity inherent in the layered construction of these networks. There has been no disruption of usage related to the proliferation of wireless networks: on the contrary, these networks are today mainly used as a local network for individuals to distribute access to the Internet.

However, examples of new uses mentioned in MANET, such as mesh networks, are almost non-existent in current use; the few initiatives are the work of research groups, wishing to experiment the behavior of protocols on localized, who take advantage of the possibilities of certain embedded hardware, such as some routers using Linux as an operating system to integrate routing functions into an 802.11 access point, and thus creating a community network, like the Citizen network in Belgium[1]. However, from the very confession of the users, these networks are rather inefficient, and pose problems ranging from the setting up of new nodes, addressing, routing, naming, reaction times, security transferred data.

So there is a problem with the use of wireless networks: why limit ourselves to the traditional mode of the client of an access point? We believe that the answer to this question lies in the current implementation of 802.11 wireless networks in our current information systems. Indeed, inheritance building network layers, and accounting with IP has led to complexity in setting up these networks.

In order to better understand this phenomenon, we will first study the approach adopted in current information systems and the relative functioning of the various elements involved in the various layers in wireless networks. Then, we will discuss new approaches, based on concepts calling into question the structuring layer; finally, through related work on wireless network security, we discuss another vision of network architecture.

1. Introduction

1.1 Traditional approach in operating systems

Operating systems more or less follow the layered model from the Internet. In order to monitor hardware evolutions while guaranteeing a stable set of programmatic interfaces (API), operating systems (or Operating System, OS) use various abstractions. The best known of these abstractions is that concerning network communication: This is the TCP / IP layer.

1.1.1 Layered operation

The TCP / IP stack is articulated around 5 main layers (see Figure .1):

- An application layer, managed by the application programmer, considered as the end user from the point of view of the operating system;

- A host-to-host link layer: this is where the actual connections and the flow control are done, by protocols like TCP.

- An internet layer or interconnection: which defines the IP addresses, and where the routing of the IP packets is carried out?

- A network access layer, which provides a physical means, as well as a medium access technique for broadcasting packets from the Internet layer, and finally a link layer. It is an abstraction with respect to the physical principles and the method of access to it.

This stack is de facto the current organization of IP-based communication systems, theoretical models of the OSI type that have not received any implementations, except in the context of protocols implemented in telecom. The most well-known TCP / IP implementation, that of BSD, is found in various current operating systems: Windows, OSX, FreeBSD, that of Linux being on a very similar principle.

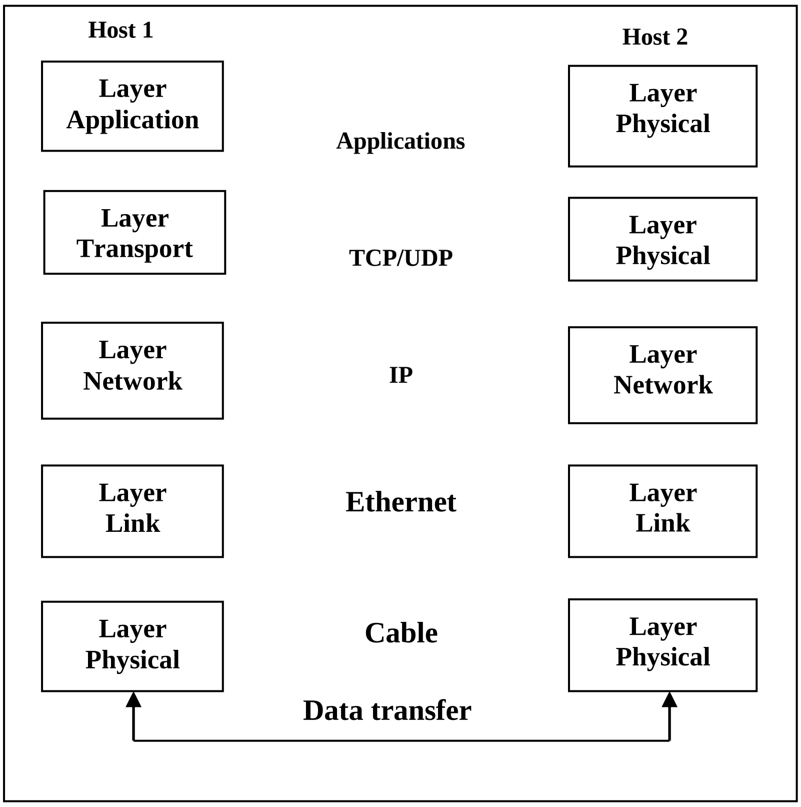


Figure .1 Simplified TCP / IP Model.

Operating systems are most often organized according to the hardware architecture on which they operate; these architectures present, for a large part, a domain segmentation (rings, or rings), which allows to introduce security within the operating system. This concept, introduced by Multicast, is still relevant in today's systems.

Therefore, the TCP / IP protocol stack is integrated in this type of security architecture: while the application layer is accessible by the users, the other layers, namely the network access, the interconnection and the link layer are executed in privileged kernel space. This security has consequences: any data exchange between the kernel and the user space can only be performed within a certain number of cycles, between 500 and 1000 cycles on average.

This particular architecture explains the slight modification of the protocol layers over the years: indeed, since the application layer is isolated in user space, new applications can be executed without calling into question the stability of the operating system; on the other hand, on most proprietary systems, it is impossible to program in kernel space; therefore, the architectural changes are isolated to the application space.

However, the progress of open source operating systems such as FreeBSD or Linux has contributed to the improvement of the intermediate layers as the years go by: many works offer interesting prospects for evolution.

1.1.2 Physical layer and MAC layer

According to the designers of Free BSD (see André Oppermann's technical publications), the designers have very little influence on the physical layers. As we have seen, the physical flows are improved by new modulations, or new physical transmission techniques (MIMO for example):

Most of these operations are entrusted to signal processing processors that perform the decoding functions. The bit streams obtained are stored in the buffers of the network card.

The MAC layer, for the same reasons, is not accessible from the operating system. MAC layer implementations are either hardware-based, with non-reprogrammable behavior, for example, cards that do not use firmware, such as Atheros cards that rely on an EEPROM; either in software form, in a firmware (or firmware) loaded by the operating system at the initialization of the card, as with Prism cards or Intel cards of the Centrino platform.

1.1.3 802.11 protocol and data link layer

On the other hand, the 802.11 protocol has meanwhile been the subject of a progressive arrival in the kernel space of the operating system; Initially, the first wireless cards were planned as an ad hoc replacement of Ethernet cards, like the first Waveland cards: the firmware of these first cards did much of the required work and provided the system with the first consolidated layer: the link layer (or data link layer).

The set of systems has more or less aligned with the format of an Ethernet frame. When a frame is received by the wireless card, it is transferred to the main memory of the system. At this point, the CPU is able to process this data. This process is called direct memory access (DMA) where the network card writes the received frame to a predetermined location in the main memory. This information flows through the PCI bus, which at the moment is not yet a bottleneck given the modulations offered by the 802.11 standard.

The package will be examined several times, including by each layer of the TCP / IP model. That is why modern operating systems use advanced caching techniques (Cache Prefetch) to place the packet in the cache as soon as it is received.

To reduce costs and memory requirements on wireless cards, vendors migrated 802.11 protocol management to the kernel, adding an additional layer to packet processing. While most of the drivers and operating systems exploit this kernel zone decoding, some drivers transfer the decoding to the user part, thereby multiplying the memory exchanges between the user and kernel zones and thus slowing down the processing of the data. .

Thus, the card market has a great diversity but similar functions:

- Intel and Broadcom cards use programmable firmware and an open source driver; the state machine 802.11 being minimal and placed in a kernel module;

- adheres cards use an euro with a MAC layer with fixed behavior. Only certain aspects relating to parameters of certain amendments (like 802.11e) are adjustable. Most of the processing is done by a kernel module, both for packet processing and the 802.11 state machines; in particular, rate control is performed by the host system.

- The USB sticks use a firmware for the management of the MAC layer, then a driver with an 802.11 state machine.

Once the 802.11 decoding is done, the abstractions used for the Ethernet cards are reused: the kernel work can begin.

1.1.4 IP layer

The kernel performs checks on the integrity of the packet, then submits the destination of the host, or if it must be retransmitted according to a routing table; if the routing is inactive, an ICMP packet is sent signaling the failure.

The route of the routing table uses hashing techniques. Again, developers are trying to optimize performance by placing these tables in the cache. The management of the routing, that is to say, the addition, the suppression the modification of the rules is carried out by a routing daemon placed in user space. The reactivity of it is necessarily reduced.

The non-local packets are thus retransmitted according to a hash table, and then sent back to a network interface associated with the route, which may be the wireless card or a card. Therefore, the packet must be transferred to the card, causing access to the system memory by the network card. The local packets, when to them, are then associated with their respective transport layer.

1.1.5 Transport layer

Local packets are assigned to the socket dedicated to their protocol - TCP or UDP. The transport layer checks the integrity of the message and determines whether the receiving socket exists in the system. If it is not, an ICMP error message is returned to the source. For TCP packets, called segments at this point, the kernel needs to examine all active TCP sessions and listening sockets. Again, a hash table is used to retrieve the destination socket.

TCP manages the orderly arrival of packets and error handling. It therefore maintains a reassembly queue and uses TCP ACK packets to retransmit missing packets or not. Linked lists are used, but new techniques appear and prefer the use of blocks to simplify the manipulation of segments.

Ethernet cards have introduced physical devices for the calculation and reassembly of segments (so-called "offload" techniques), but, from a system point of view, these techniques are too generic and call into question the network architecture too much. Current nuclei. To our knowledge, only wired Ethernet cards exploit these physical techniques. From then on, the packages are available for user space applications.

1.1.6 Application layer

The main task is to transfer the data from the kernel space to the user space. The applications therefore use, depending on the transport layer, socket-type system calls, then listen, accept for TCP-based connections.

Depending on the degree of optimization, the application can use select or poll functions to check for the presence of data on a socket, and other mechanisms can be interesting in a massively multi-process application (such as Web server).

2. Challenge of layered architecture

The layered system has enabled a formidable deployment of networked applications, as well as a strong competitive development of adapters. In fact, builders and application developers do not need to know each other's business. But, as we have seen, in a wireless networks perspective, these layers are considerably multiplied: appearance of the 802.11 protocol, various hardware integration solutions...

Therefore, for the new uses of wireless networks, this architecture appears inadequate for 3 essential reasons: first, the wireless links pose specific problems to this technology; secondly, the creation of links is opportunistic by radio communication; and finally, the communication on this medium is by definition very different.

First of all, the wireless link inherently creates new problems for protocol designers, as we studied in our 802.11 autopsy: the software package formed by the protocol stack does not allow the simplified resolution of these protocols. problems. The classic case of TCP is the perfect example: a physical error can cause the transport layer to react to a congestion problem that is not a problem.

On the other hand, wireless networks offer opportunities for opportunistic communication that cannot be used in traditional layered design. In particular, radio links vary over time, and this temporal adaptation can make the object of an adaptation of the transmission parameters, which we have seen, is little used in the implementations: the correcting codes used in the OFDM modulations are for example inaccessible to the MAC layer: the adaptation mechanisms that we We have mentioned that this is based on the packet error rate, and not on physical indices that would allow rapid adaptation to the medium.

Similarly, physical layers can offer simultaneous (in the statistical sense of the term) reception of several packets at the same time, and the very nature of radio broadcast is not used by the protocol layer (the first step of a packet). card being to identify if the package is intended for him or not). Therefore, the routing mechanisms proposed for example in MANET, even if they try to approach the reality of the medium by the use of suitable metrics for example, do not finally take into account the radio topology: the retransmission mechanisms do not take into account temporal variations of the channel, and therefore, between the reception and the retransmission of a packet, too many elements could fluctuate.

The optimizations proposed by the researchers, studied during our previous autopsies, represent localized violations of layer architectures. Since then, part of the community has explored the so-called inter-layer architecture track, aiming to overcome the layered model in the face of the challenges posed by wireless networks.

3. Inter-layer approaches: architectures and frameworks

Several authors have proposed an inter-layered view of the network architecture. Two complementary contributions provide a comprehensive analysis of the possibilities and knowledge of an inter-layer architecture, both positive and negative aspects. In the first [88], the authors Kawadia and Kumar propose a precautionary approach to inter-layer architectures.

One of their main concerns is the design in spaghetti code: each optimization can be isolated and lead to non-reusability. Therefore, if wireless networks are to become innovative architectures, they must maintain strong founding principles, such as computer architectures, heir to the architecture of Von Neumann, which for example distinguishes well the memory, the control unit , arithmetic and logical units. Similarly, the authors cite the TCP / IP model as an example: as we have seen, layer separation was once essential for the large-scale development of network interconnection.

However, as noted above, the de facto layered model for wireless networks is not necessarily the most appropriate, especially because of the new paradigms of wireless communication.

The authors thus resume the basics of digital wireless communication. Thus, the construction of this communication is based on the theory developed by Shannon, according to which the sources can be decor related from the transmission channel without loss of optimality. There is therefore no point in making a source of bit stream adapted to a given channel. Therefore, a first architectural element appears: the wireless card can therefore perform optimal channel coding for a given transmission channel for a wide variety of sources (thus digital data).

The authors therefore claim that the problem of positronization of layers is elsewhere, and more precisely is at the level of data transfer. Shannon explains that the capacity of a channel is given by its signal-to-noise ratio. In wireless, several strategies for retransmission of data are possible: the notion of link does not exist: each node emits energy that is superimposed on others and is therefore simultaneously received by the recipients. Several types of cooperation are therefore possible: a group of nodes can cancel the interference of a given group for the benefit of a third party; or rather, it can relay the signal according to different strategies: amplify and retransmit, or decode and reissue. For the authors, the new vision is not a mechanical displacement of information, but rather an electronic transfer of it.

The authors therefore justify in this publication two fundamental elements of wireless networks:

- Multihopping is optimal for a given order of magnitude: in the framework of the traditional principle of decoding and retransmission, this operation is optimal when it is performed from node to node as long as the load of the nodes can be balanced by a multichannel routing.

- Other strategies are optimal if the attenuation conditions are low: the relaying with suppression of interference can be more interesting.

Therefore, these results are interesting. They justify at the architecture level the presence of the decoding / retransmission operation in the most common case. In addition, they also justify the need for path routing, and therefore the existence of this notion in wireless architectures. Finally, the existence of a transport protocol,

Permitting the establishment of a point-to-point pipe is therefore justified. The authors find much of the current principles of networks and therefore by extension of current wireless networks. However, their approach shows that any inter-layer optimization approach can only result in bandwidth optimization, and not infinite improvement even in networks with large numbers of nodes.

The consequences of the first conclusions of this publication are interesting: we should not expect wireless networks a fundamental increment of the network capacity by the presence of multiple nodes; at best, one can, through the use of inter-layer techniques, see an optimization of the network capacity used and approach the theoretical maximum capacity. We must therefore leave the paradigm of wired networks introduced by Ethernet: the addition of a switch increasing the capacity of the networks must therefore disappear in the minds of the designers of wireless networks. The addition of new nodes does not result in an increase in bandwidth, but rather in an increase in connectivity and the effectiveness of this connectivity can only be improved by the use of inter-layer system.

3.1 Theoretical inter-layer architectures

In the continuity of this first reflection, the authors of [70] propose a study of cross-layer architectures allowing to reach these new objectives of performance, thus completing the contribution studied previously. Srivastava and Motani list the different types of propositions, illustrated in the diagram of Figure.2.

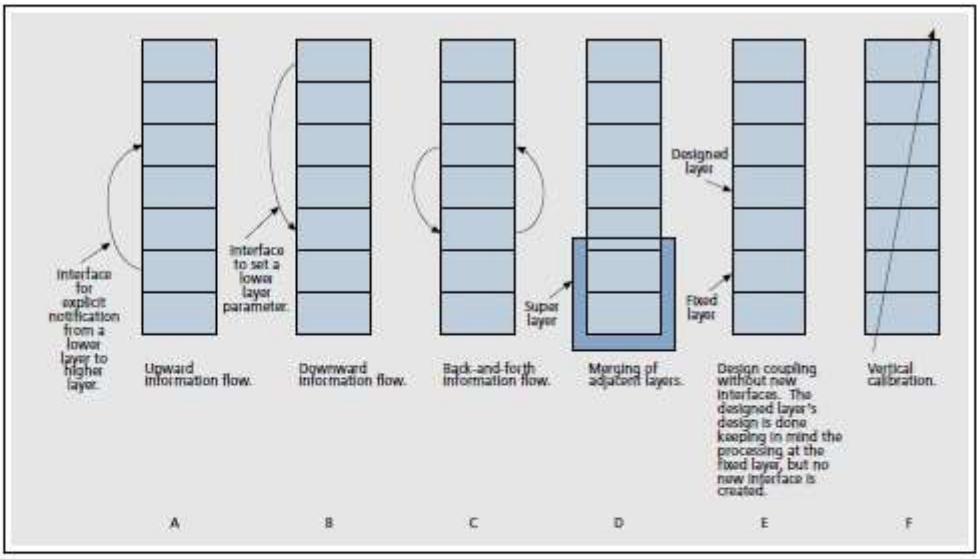

Interface for explicit notification from a lower layer to higher layer.
Upward information flow.
Interface to set a lower layer parameter.
Downward information flow.
Back-and-forth information flow.
Super layer
Merging of adjacent layers.
Designed layer
Fixed layer
Design coupling without new interfaces. The designed layer's design is done keeping in mind the processing at the fixed layer, but no new interface is created.
Vertical calibration.
A
B
C
D
E
F

Figure.2 Types of crosslayer architectures observed in the literature. (From the publication of Srivastava et al.)

The first architecture presented consists of reassembling information from the lower layers to the upper layers. This type of architecture is used in several real cases: notably by the mechanisms of adaptation of the flows [9] [39] [52] [84] [19], where the information of the physical layer, like the rate of packet error is used to determine if the physical modulation needs to be adapted. But this type of architecture also concerns the higher layers: a router can notify the transport layer of an explicit congestion by the use of the ECN bit, added in the specifications of TCP [78].

A second architecture consists in using reverse communication, layers higher than lower layers. Typically, this architecture is implemented indirectly in the WiFi standard 802.11e amendment: the four queues offered (Best Effort, Background, Video, Voice) are filled according to the Type Of Service (TOS) field of the package IP, as proposed by Park and Choi [98]. Thus, by a simple indication of the upper layers, the action of the lower layers can be modified.

These two architectures can be combined, for example in advanced scheduling techniques that take into account the traffic to be transmitted (upper MAC layer) and power emission constraints (lower physical layer), as proposed in [87].]: thanks to mechanisms of intercommunication between the two layers, the authors manage to reduce interferences not reducible by the power control by a more efficient scheduling of the users, which takes into account the constraints of power of each one.

Other architectures are more simplifying: for example, the creation of a super-layer by merging two layers. Although the works do not cite this superlayer as such, most work on consistent optimizations between the physical layer and the MAC layer indirectly leads to this type of architecture. Another solution is also to develop a layer by anticipating the operation of a lower layer. In particular, this kind of architectural solution is used for application layers: the lack of access to the lower layers therefore requires adaptive mechanisms at the application level. The use of the RTCP as a feedback loop in multimedia networks using the RTP protocol is its practical realization [73].

The last solution mentioned by the authors is a form of vertical calibration: for example, the authors of [92] present an architecture where the adaptation to the required delay modifies the error correction mechanism (ARQ) and also the correction codes of physical error. Thus, the architecture makes it possible to preserve the choices in terms of delay by acting on all the layers.

The multiplicity of these architectures leads to a multiplicity of solutions to implement them in the systems. Srivastava and Motani establish three types of achievements (see Figure 3):

- A direct communication between the layers: the operating variables of a layer are accessible by another layer, instead of being internal and restricted to a given layer. However, other types of communication may be considered: for example, the use of the headers of each protocol may allow

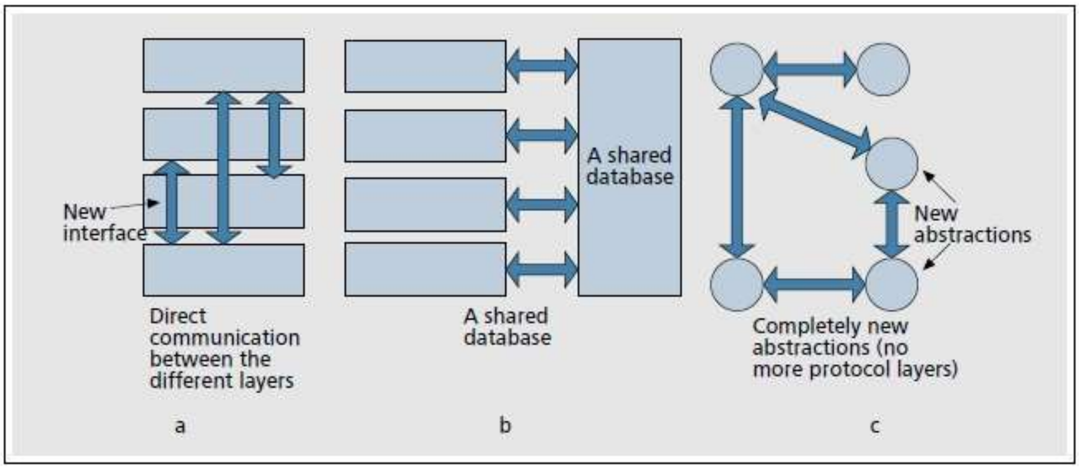

New interface
Direct communication between the different layers
a
A shared database
A shared database
b
New abstractions
Completely new abstractions (no more protocol layers)
c

Figure .3 Types of Cross layer Implementation Proposals. (From the publication of Srivastava et al.)

Communication between the layers. For example, CLASS provides a restricted set of shortcuts between layers. Clearly, this type of communication is implementable in the current kernels, since the TCP / IP implementation directly uses the same memory buffer for all the layers of a packet. One can thus imagine an interlayer mechanism giving access to all the layers in the kernel of the operating systems. However, this solution is exposed to the problem of management of the memory zones between supervisor mode and user mode, and therefore the inter-layer mechanisms can be underperforming or pose security problems.

- Other proposals evoke the possibility of a common database and informed by all the actors of the layers. Obviously, this transversal vision applies very well to interactions involving all layers. However, the heaviness of such a mechanism can cause problems in an embedded and constrained environment as the core of an operating system.

- Finally, there remains the possibility of creating new abstractions: that is, going beyond the layered model, the concepts of encapsulation and proposing new visions. These 3 types of architecture are found in the frameworks inter-layers resulting from the experimentation.

3.2 Frameworks resulting from experimentation

Experiments on wireless networks have resulted in a large number of frameworks dedicated to wireless networks.

One of the first experimentation frameworks on networks is proposed by Morris in 2000 [69] and its wireless extension by Bicket in 2005 [72]. This framework was used to develop Roof net, an unplanned mesh network located in Cambridge, Massachusetts. Each Roofnet node is a standard silent PC with an 802.11b card and an unidirectional antenna. The cards use an ad hoc mode derivative of 802.11b, and do not use virtual carrier sensing (no RTS / CTS). This derived mode does not send Beacon, eliminating the parsing phenomenon observed in networks using the ad hoc mode (raised by the Atheros chipset developer developers, MadWiFi).

The software is organized as follows: Click provides a user mode routing architecture, based on the juxtaposition of elementary elements. The model is object-oriented: each element of an element class has one input, several outputs, and one configuration input. Two modes of communication are implemented: push and pull. In push mode, packets sent to an item are processed; in pull, it is the element that solicits the entry of packets. Packet processing can rely on queues that are elements like the others. This formalism is very powerful because it makes it possible to get rid of the layer mechanism by isolating the manipulation of it at the level of each element and reduces the study to the construction of a routing scheme. The interaction with the kernel space is achieved through virtual interfaces.

As part of Roofnet, each 802.11 card is used as the final input and output of packets. The addressing is determined by the unique Ethernet address of each card, from which the last bits of the IP address are derived. Each node is equipped with an Ethernet interface for the wired connection to the Internet. If such a connection exists, the node then offers a gateway to the Internet. When forwarding packets to a Roofnet node, Click's configuration uses a metric based on the largest bandwidth between the sender and receiver nodes. In [72], the authors evaluated the influence of the metrics based on the quality of the link, collected during of the route request, the protocol used being a derivative of DSR. The metric used is ETT, derived from ETX. ETT predicts the total transmission time of a packet along a route. This metric is built around short and long beacons transmitted regularly. Since this metric relies heavily on the modulation used on each link, the authors have modified the rate selection algorithm by replacing ARF with Sample Rate, which adjusts the rate based on probe packets sent at regular intervals.

The observed performance is similar to that of other mesh networks, yet the construction of it is greatly facilitated by the original approach of the modular router Click.

Biswas and Morris [97] also used Click to implement ExOR, a MAC layer and routing protocol that uses opportunistic conditions at each hop to improve total bandwidth. The operating principle is oppor -Tunist: Each radio-based node determines if it is closest to the destination, and returns the packet in turn. ExOR is based on the ETX metric, and is built on Click for its implementation.

Other authors have successfully used Click for their experiments.

Stoica [94] used Click to build a virtual MAC layer overlaying the usual MAC layers. This abstraction layer makes it easier to test new protocols for the MAC layer. The abstraction layer is based on the capabilities of the cards: transmission, reception, possible activation of the RTS / CTS, but more importantly, the fine management of the queues. Indeed, this overlay is based on the ability to modify the content of card queues. This approach is interesting, however we can question the granularity and realism of such a MAC layer, knowing that the intrinsic mechanisms of access to the channel are not easily manipulated. For example, using the Binary Exponential Backoff is not disengaged on this prototype. Be that as it may, the authors obtain good results in terms of equity, especially on the TCP streams.

Another example of use is Brewer's WildNET [83], which uses Click as part of long-distance WiFi networks. In this area, the usual features of 802.11 are called into question: first, the packet transmission mechanism based on an acknowledgment is subject to propagation phenomena: long distance, the transmitter must wait longer for the arrival of the acquittal. The use of the channel is reduced. In addition, if the delay of receipt of the acknowledgment is greater than the standard expiration, the packet is considered lost by the issuer who will then reissue it. Second, the CSMA / CA access mechanism is also disrupted. For the standard 802.11 DIFS, ie 50 μs, the distance for which collision avoidance is valid is 15km. Third and last point, interference and more specifically other 802.11 networks reduce the efficiency of transmissions.

To overcome these problems, the authors have proposed a replacement of the CSMA / CA by a TDMA type mechanism, with the addition of burst acknowledgment, allowing the acknowledgment of a number of packets with a single acknowledgment. To implement these proposals, the authors used Click to build the new MAC layer, modifying the driver to disable the acknowledgments, and the CCA (which allows immediate transmission on the channel without delay). WildNET authors could not use Click's time management mechanisms because the granularity of the time slots is insufficient for the kernel. Moreover, between the moment when Click's interface sends a packet and its actual send, a difference is noticed. However, this difference being limited, the authors were able to eliminate this pitfall by taking into account these differences in the calculation of the TDMA allocation.

In the same spirit, XORP [89] allows the implementation of new routing protocols. XORP is composed of a packet transmission engine (which can be realized as a Click module) and high-level applications in user mode, allowing the implementation of new protocols. XORP is mainly dedicated to the routing problem.

Other experimental frameworks have been developed for cross-layer research in wireless networks. Aache et al. have proposed XIAN [79], a framework to gather in the form of an API all the statistics of a wireless card, in this case Atheros cards with the MadWiFi driver. This API is then implemented to perform a routing taking into account quality of service constraints. Xian is built around a kernel module that retrieves statistical data from the hardware driver and exports it to a user-mode library. Xian on the other hand does not allow to modify the access parameters to the channel of the card, that it is physical or MAC. This is more of a layer instrumentation in this case, but the chosen architecture leaves the door open for future modifications.

Ben Adesslem et al. have proposed Prawn [90], a tool for rapid communication protocol implementation over 802.11 networks. The aim of the authors is not here to propose a powerful architecture, but to provide a means of testing

a protocol, unlike Click or XORP which offers the possibility of a powerful implementation of a protocol. The environment consists of two elements, the engine, or Prawn Engine, consisting of elementary blocks over which protocols are built. Then, the Prawn library offers an API allowing transparent access to the underlying elemental bricks. The goal here is to focus on the upper layers, ie the IP level. These frameworks are a first step towards a new vision of the network architecture model. Beyond the traditional layered model of the TCP / IP stack, they offer facilities for experimenting new optimization methods, either by building elementary bricks (Click or Prawn) or by exposing internal data. to the whole system (XIAN). Therefore, a question arises: can we go further in a new vision?

3.3 Towards a new vision of radio interfaces

The frameworks that we have just studied in the previous section basically answer to needs in term of "network" oriented vision, ie that the layered model is little revisited.

However, in the field of wireless networks, researchers from other fields have worked to develop techniques for advanced analysis and defect detection: on the one hand, research in safety, and on the other hand, researchers working on software radios.

3.3.1 Another vision: contributions from security

Researchers quickly became interested in the challenges posed by wireless networks. In fact, unlike wired networks that are not very sensitive to information leakage through the medium, poorly configured wireless networks can constitute an unauthorized entry point into a network. The physical confinement by the management of the power of emission for example is a false solution because an attacker can always use aerials (antennas, amplifier) with high gain allowing the listening and the emission of data on the frequency of operation of the network.

Very quickly security researchers have sought to assess the security of wireless networks. The domain first focused on the security of the security protocol provided by 802.11, WEP (Wireless Equivalent Privacy) and then WPA, supposed to provide a way to secure 802.11 data encryption.

The advanced theoretical study of this protocol since 2001 by Borisov et al. [93] reveals significant failures of cryptographic functions in the 802.11 protocol. Indeed, the passive listening of the traffic of an 802.11 network allows the recovery of the common session key used between 2 802.11 devices using encrypted 802.11 messages using WEP. As a result, 802.11 fault discovery announcements follow one another ([91], [86]), and the community is looking for techniques to demonstrate 802.11 session key (Proof-of-concept, PoC) rapid recovery.

It is then that a significant number of tools dedicated to the examination of the security of the networks appear. First, a number of scanners, which lists the beacons received by the onboard 802.11 equipment, either passively (by listening management packets) or actively (by sending 802.11 packet type Probe Request). WEP research encourages the emergence of new attack techniques, based on common security concepts, such as the ARP packet replay [85], in order to increase the number of encrypted packets transmitted and thus to discover more quickly the session key. So researchers are looking to make their own packets and inject them on a given frequency, using the possibilities of some wireless chipsets (Atheros, Zydas, Prism can arbitrarily send a packet from an interface configured in monitor mode for example).

At the same time, there are other problems affecting wireless networks: sending certain types of packets causes some cards to crash their kernel-based drivers, and therefore also drive the kernel of the operating system. Maynor, for example, demonstrated with his "Hijacking a MacBook in 60 seconds or less" attack the vulnerability of certain drivers to certain types of corrupted 802.11 packets, which resulted in the possibility of remote control of the machine. To highlight these security flaws, some researchers have proposed Fuzzing techniques [77], [81]. These techniques consist of submitting to a set (map without

wire, driver, operating system) a set of malformed packets to test the robustness of the set.

All of these two needs (packet generation of a given type, or malformed packet generation) made it necessary to use frameworks dedicated to the creation and injection of packets. To date, there are several relatively generic types of abstraction:

- Several libraries allow the creation of 802.11 package of any type:

Airbase2, Lorcon3. These libraries allow the reuse of the same code on several types of wireless cards.

- other frameworks are more generic: Scapy is a set of Python language lines allowing the arbitrary creation of any type of packet, 802.11, ARP, IP, TCP, as well as the creation of a new protocol. Packet Forge is a similar set but in C and is rather dedicated to BSD systems. They both allow the recording of new protocols.

The study of Scapy, a tool developed by Philippe Biondi is interesting. Indeed, it relies on the Python language to facilitate prototyping of package manipulation. Scapy therefore works in user space, and uses socket open on the interfaces for the recovery and injection of traffic. The processing is therefore done entirely in user space, which avoids the exchange of data between the user space and the kernel, greedy in terms of management for the operating system. Scapy uses a very simple formalism for defining a new type of package. The authors of [74] have used it successfully to implement new tools dedicated to Enterprise Service Bus security. The declaration of the data structures used by the protocol is as follows:

```
class Modbus (Packet) :
        name = "Modbus"
        fields_desc = [Short Field ("transaction_id", 0),
        Short Field ("protocol_id", 0),
        Short Field ("data_length", none),
        Byte Field ("unit_id", 0),
        Byte Field ("function code", 0),
       Data Field ("data", {})]

def posts_ build (self, p, pay):
   if   self .data length is None:
    length = len (p [8 :]) +2
    p = p [: 4] + struct .pack (' > H', length) + p [6]:
     return p
```

The class defines the type of packet, the format of the different fields, the actions to perform (checksum calculation) after building a packet. Scapy already integrates all 802.11, IP, TCP and UDP protocols. It is therefore widely used for the implementation of tools dedicated to the study of security. Some derived applications

Scapy, such as WiFiTap, offers in some lines of code the ability to communicate over a wireless channel without the heavy 802.11 protocol.

These tools are interesting in more than one way: not only do they offer quite different abstractions from the frameworks resulting from the research, in particular through their toolbox vision of the traditional network, but in addition, these tools are used in an operational way, from so that they are commonly used in terms of proof of concept. Indirectly, these tools challenge the layered model by allowing direct access to certain layers (802.11, IP etc ...) directly from the user space, where usually only the routing daemons are tolerated. These tools therefore contribute to dissociating the network layer from its usual integration in the operating system, and in that they join some cross-layer visions proposed by network researchers. However, as we have seen, these tools even though they approach the lower layers of the network are limited to handling the 802.11 protocol. Is it possible to manipulate the physical layer as easily?

3.3.2 The new possibilities of software radios

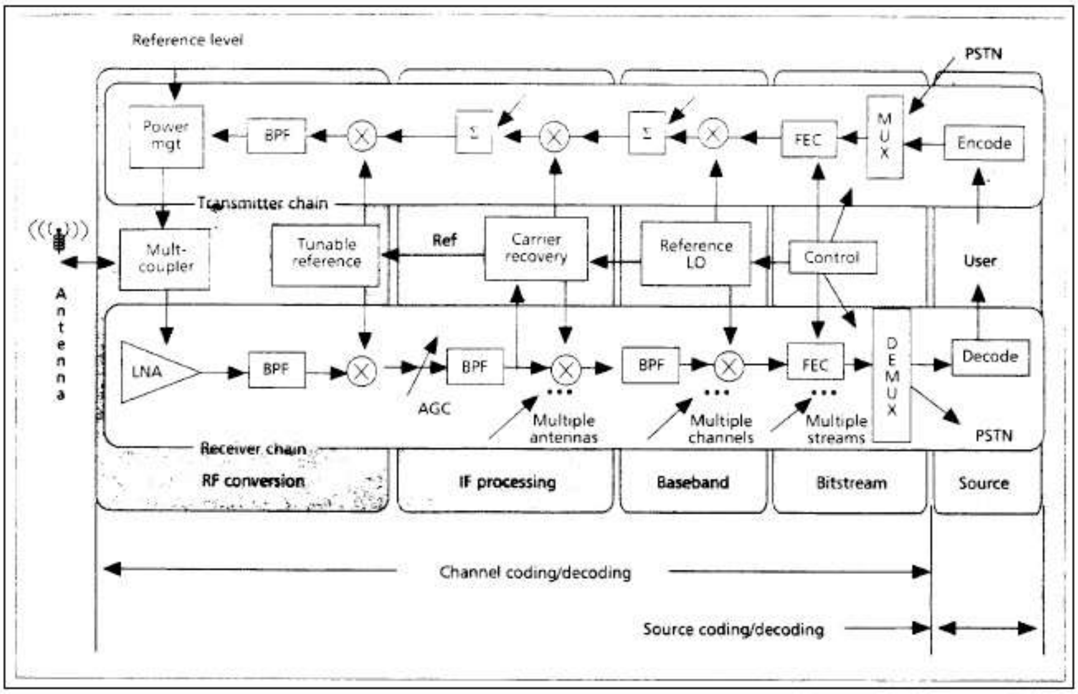


Figure.4 Architecture of a software radio (from "The software radio architecture" by mitola et al)

To answer the last question raised in our previous section, we need to reposition our discourse in light of the latest advances in digital transmission. As we saw in our chapter dedicated to the study of the 802.11 protocol, the physical parts of the 802.11 devices consist of components dedicated to digital signal processing (mixers, filters, amplifiers, modulators, demodulators, FFT ...). The idea of Software Defined Radio (SDR) is not new: as early as the 1990s, Mitola proposes a software radio architecture in [80]. Mitola defines the main elements of software radios (see Figure.4): a power supply, an antenna, a multi-band radio frequency converter, and analog / digital converters with a processor and memory that perform the functions previously performed by dedicated components. . Mitola also specifies the ideal placement of each element, and assesses the cost in terms of operation and the required timings. From his analysis, we retain the need for real-time operation (see Figure.5) and the requirements in terms of millions of operations per second (see Figure.6).

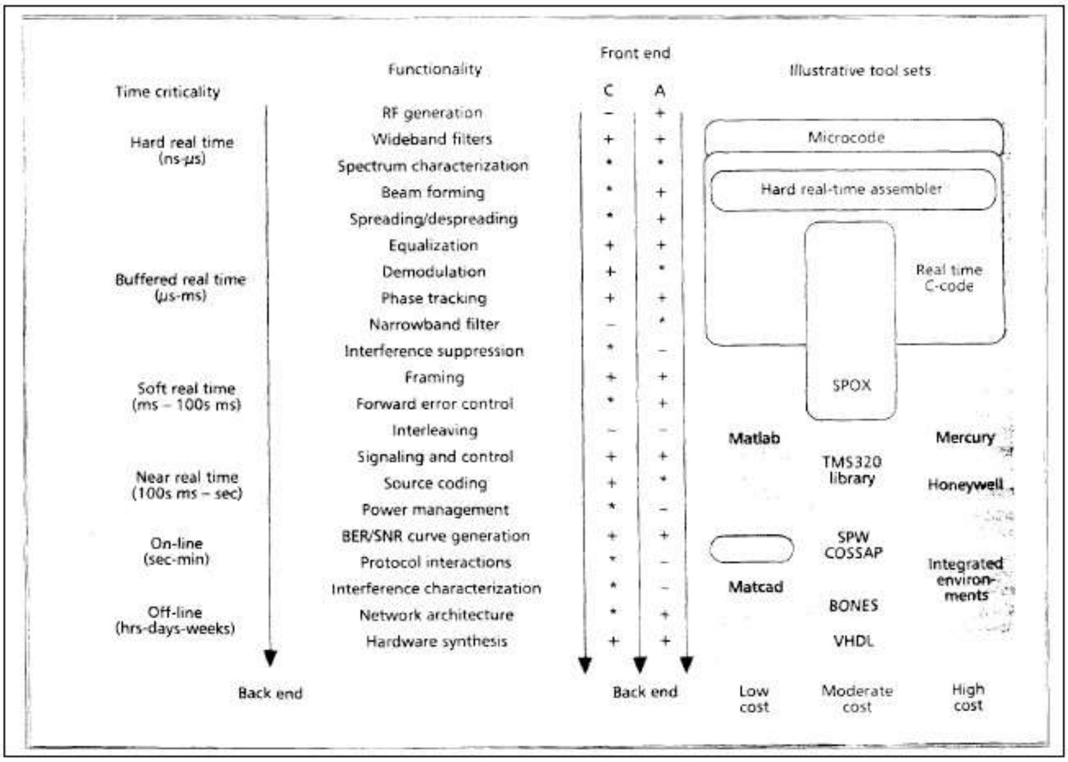
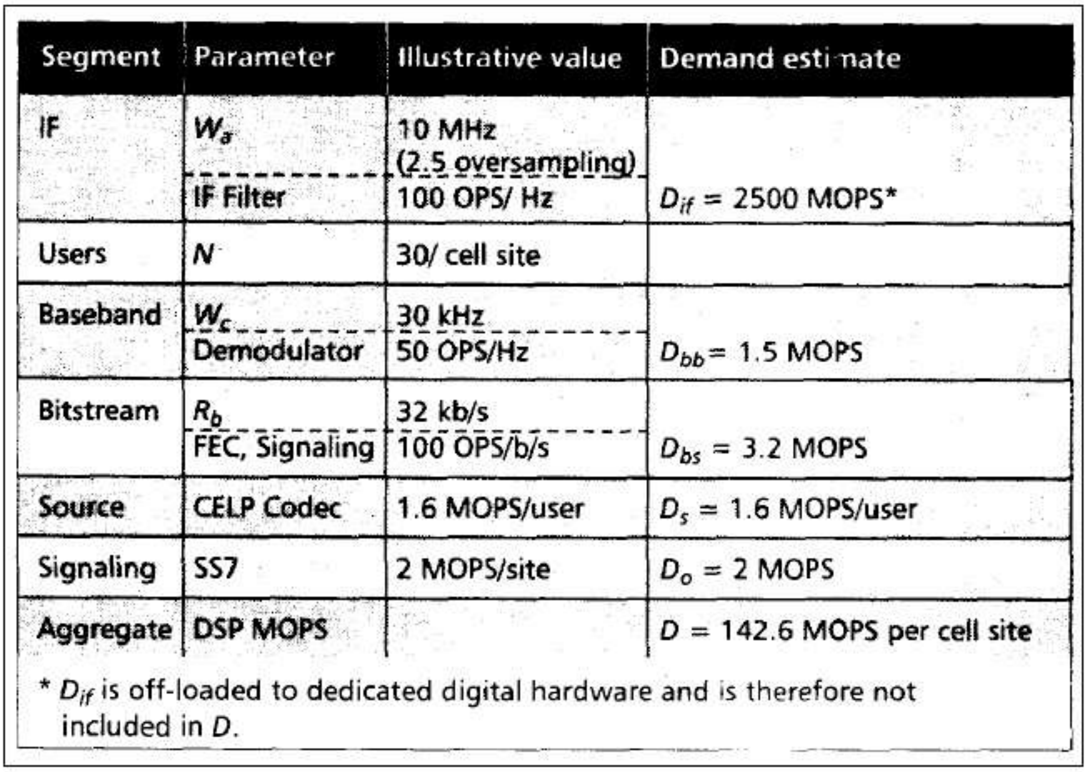


Figure .5 Time constraints of integration of a software radio (extract of "The software radio architecture" by Mitola et al).



Segment	Parameter	Illustrative value	Demand esti nate
IF	W_a	10 MHz (2.5 oversampling)	
	IF Filter	100 OPS/ Hz	D_{if} = 2500 MOPS*
Users	N	30/ cell site	
Baseband	W_c	30 kHz	
	Demodulator	50 OPS/Hz	D_{bb}= 1.5 MOPS
Bitstream	R_b	32 kb/s	
	FEC, Signaling	100 OPS/b/s	D_{bs} = 3.2 MOPS
Source	CELP Codec	1.6 MOPS/user	D_s = 1.6 MOPS/user
Signaling	SS7	2 MOPS/site	D_o = 2 MOPS
Aggregate	DSP MOPS		D = 142.6 MOPS per cell site

* D_{if} is off-loaded to dedicated digital hardware and is therefore not included in D.

Figure.6 Constraints in terms of operation numbers per second for a software radio (from "The software radio architecture" by Mitola et al).

From this study, we retain the questions raised by Figure .5: the temporal constraints are different for all elements of the chain of a software radio. We distinguish at the physical level critical operations at the level of RF signal generation, then constraints of the order of nanoseconds / microseconds at the level of signal processing, then constraints of the order of milliseconds at the level of the signal. error correction, then the second at the protocol level. However, these constraints have been established for lower frequencies than our wireless networks: in the Gigahertz range. For 802.11 networks, transmitted symbols are of the order of one microsecond (4 microseconds for an OFDM symbol). as for the protocol messages, the times IFS are of the order of the multiple of the slot, that is to say 9 microseconds in 802.11a or 802.11g. Therefore, these temporal considerations are further exacerbated by the microwave communications involved in 802.11.

This issue of time constraints is also present in other projects concerning radio communications. The Open Air Interface [96] initiative, which aims to provide an open platform for experimentation in digital radio communications, is composed of several subsets, some of which are dedicated to solving real-time problems (for example). software radio and embedded systems using real-time operating systems). At the same time, the project also includes work on access methods (cellular and mesh topologies, interlayer scheduling, distributed resource control) and wireless networking (Mobility Management, routing protocols adapted in mesh mode and cellular). The integration of these different works from the design stage, notably through the use of physical layers simulated by elements operating in real time, marks a definite change in the design of new architectures.

The software radios involving the 802.11 protocol with complete functionalities are thus at the moment rather difficult to implement. However, some projects are at the time of this writing well advanced. The GNU Software Radio project offers a fairly complete framework for software radio programming.

For the hardware part, it relies on a set of components named USRP, which several researchers have evaluated performance, including Dhar et al. [95]. For the moment, it seems that this architecture does not allow to achieve equivalent rates to integrated cards, especially because of the hardware architecture that does not support a real-time operation. However, with only reasonable data rates and low operating frequencies, the researchers were able to integrate this software radio and use it as a physical layer and MAC layer for the Click routing frameworks we studied previously. . In a single approach, researchers have thus managed to consider the elements of the physical layer as usual Click elements, allowing great flexibility for the design of these new radios.

Dhar et al. also evoke in their article the possibilities in term of interlayer approach: the memory space being the same between the physical radio components and the components of the higher layers, it becomes very easy to envisage exchanges of information between the different layers, as suggested in Figure.7. The complex nature of the operations to be performed to ensure the usual functions in radio frequency that they are performed in user space, as any other application.

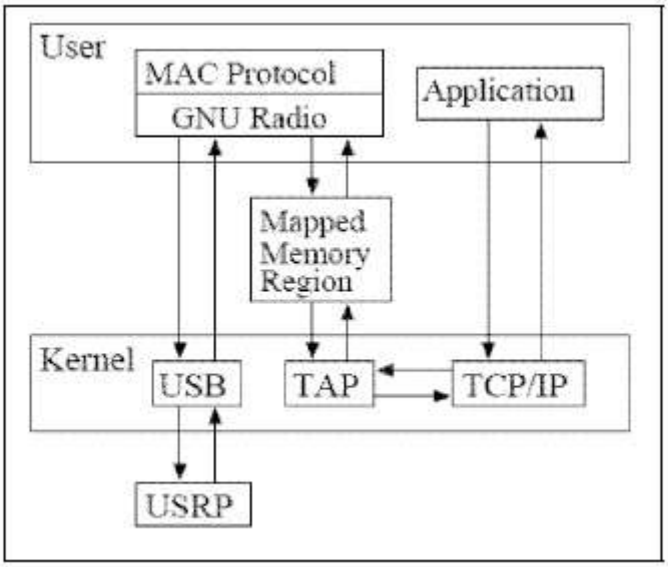


Figure .7 Architecture of a complete software radio integrating physical part (GNURadio) and higher layers (Click). (From the publication by Dhar et al.)

4. Conclusions

From this study of the current architecture of the operating systems, we were able to easily identify the reasons for the success of the layered architecture. Indeed, it has made it possible to isolate each problem (physical access, management of multiple access, packet routing, establishment of a transmission channel) in order to provide connectivity functions to the applications. However, like our previous autopsy of

802.11 wireless networks have clearly shown, this layered model is questioned when used with wireless networks, especially for performance reasons. If researchers have been able to successfully provide solutions to several performance problems (multiple flow management, rational use of the channel by multiple users, new methods of spontaneous communication in mesh networks), these solutions remain too specific and especially tend to decrease the interest of the layered model that has been an architectural success as the Internet proves.

The need to communicate all the layers of the TCP / IP model is therefore justified, particularly with regard to the nature of the solutions found. However, these solutions must be part of a new architectural model, inter-layer architectures. To date, several research studies have tried to describe this model generically, producing recommendations without reaching a consensus allowing the development of these new architectures. The contribution of the experiment made it possible to highlight the interest of works aiming at transforming the network into a series of functional blocks, like the framework Click. However, in the context of wireless networks, low layers such as the MAC layer and especially the physical layer resist this transformation, especially because of temporal considerations. Indeed, the temporal requirements are increasing as we plunge into the lower layers of the TCP / IP model.

The arrival of software radios and early experimental frameworks (such as the GNU Radio project) has highlighted these problems, and to date it is still impossible to achieve performance levels in terms of physical data rates on embedded software radios. on usual equipment. However, these limitations can be overcome by an adaptation of the communication architecture between the different components.

One of the starting points of these new architectures can be the vision resulting from the research in security: in this domain, the network is considered at the level of its most elementary entity, the package. The tools developed in this discipline offer very interesting possibilities in terms of manipulation of these packages and suddenly, offer interesting possibilities for the rapid creation of spontaneous wireless architectures. In addition, already in user space, they can integrate more easily with future software radio architectures.

It is therefore possible to rethink the architecture of the current networks by reasoning on the reaction of the system to the reception of the packets. In fact, whether at the routing, transport, session or application level, the state machine mechanisms are mainly dictated by events related to the reception of packets:

discovery of new routes by packet sending, acknowledgment at TCP level by sending of a given package, and of course, change of the application to the reception of certain packets.

This isolates the layers de facto from the IP layer: indeed, the temporal granularity of these layers is identical. At the MAC level, things are already different: the sending of a packet is subject to the statistical multiplexing mechanism, which depends on parameters related to the physical occupation of the channel, which, indirectly, links part of the physical layer and the MAC layer within the same time space. The physical layer may be isolated temporally because of the phenomena involved (duration of a symbol on the very short physical channel).

With these new bases, we can develop a new architecture that takes irreducible constraints into account while opening up the perspectives of an interlayer vision.