CROSS LAYER DESIGNS IN WLAN SYSTEMS

Edited by ProfNizar Zorba, Charalambos Skianis and Christos Verikoukis

Book Chapter

INTRODUCTION:

WHY CROSS-LAYER? ITS ADVANTAGES AND DISADVANTAGES

Authors:

Dzmitry Kliazovich

University of Luxembourg
6 rue Coudenhove Kalergi, L-1359
Luxembourg
Email: dzmitry.kliazovich@uni.lu

Fabrizio Granelli

DISI - University of Trento
Via Sommarive 14, I-38050
Trento, Italy
Email: granelli@disi.unitn.it

Abstract:

Growing interest and penetration of wireless networking technologies is underlining new challenges in the design and optimization of communication protocols. Traditionally, protocol architectures follow strict layering principles, which ensure interoperability, fast deployment, and efficient implementations. However, lack of coordination between layers limits the performance of such architectures due to the specific challenges posed by wireless nature of the transmission links.

To overcome such limitations, cross-layer design has been proposed. Its core idea is to maintain the functionalities associated to the original layers but to allow coordination, interaction and joint optimization of protocols crossing different layers.

This chapter introduces the reader with the notion of the cross-layer design, outlining motivations and requirements, presents the main building blocks enabling collaboration between layers, and compares available signaling architectures. Then, after mentioning current status of standardization activities in the field, it presents novel architectural solutions involving cross-layer design which are proposed in the framework of next generation network communications.

I. INTRODUCTION

Wireless networks represent technologies with growing interest and expectations in the world of communications. This proposed new challenges in the design of communication protocols, which are required to adapt to new features of the networking environment like shared channels, limited bandwidth, high error rates, increased latency, and mobility.

Traditionally, protocol architectures follow strict layering principles, which provide an attractive tool for designing interoperable systems for fast deployment and efficient implementation. ISO/OSI model [16] was developed to support standardization of network architectures using the layered model. A protocol at a given layer is implemented by a (software, firmware, or hardware) entity, which communicates with other entities (on other networked systems) implementing the same protocol using Protocol Data Units (PDUs).

The main advantage deriving from the layering paradigm is the modularity in protocol design, which enables interoperability and improved design of communication protocols. Moreover, a protocol within a given layer is described in terms of functionalities it offers, while implementation details and internal parameters are hidden to the remainder layers.

However, such lack of coordination among the layers limited the performance of such architectures in front of the peculiar challenges posed by wireless nature of the transmission links.

To overcome such limitations, cross-layer design was proposed. The core idea is to maintain the functionalities associated to the original layers but to allow coordination, interaction and joint optimization of protocols crossing different layers.

In Section II of this chapter we first provide an overview of existing wireless technologies - ranging from personal area to wide area networks. The main characteristics and wireless performance metrics are then discussed and summarized in a table for providing a comparison of the different wireless technology alternatives.

Finally, existing solutions for optimizing performance of communication protocols are reviewed and motivation for cross-layer design is presented.

Section III introduces the reader with the notion of the cross-layer design, provides motivation aspects and requirements of cross-layer solutions. It presents the main building blocks enabling collaboration between layers and compares available signaling architectures. The section is concluded with overview of current standardization activities and trends in cross-layering.

In Section IV we outline novel framework solutions based on the paradigm of cross-layer design, which could be included in next generation communication networks.

In Section V, conclusions are presented with a short discussion outlining pros and cons of cross-layer design and presentation of future directions in the field of cross-layering.

II. OVERVIEW OF WIRELESS NETWORKS

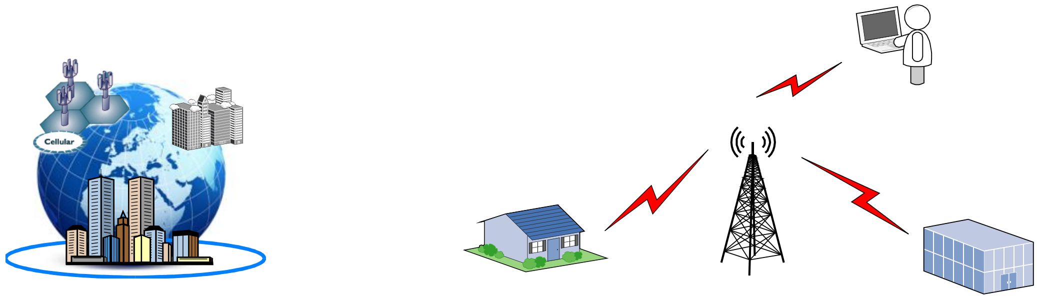
Wireless networks are becoming increasingly popular in telecommunications, especially for the provisioning of mobile access to wired network services. As a consequence, efforts have been devoted to the provisioning of reliable data delivery for a wide variety of applications over different wireless infrastructures.

In wireless networks, regardless of the location, users can access services available to wired-network users. In this scenario, the IEEE 802.11 standards represent a significant milestone in the provisioning of network connectivity for mobile users. However, the 802.11 medium access control strategy and physical variability of the transmission medium leads to limitations in terms of control over bandwidth, latency, information loss, and mobility. Moreover, the deployment of the Transmission Control Protocol (TCP) over IEEE 802.11 networks is constrained by the low reliability of the channel, node mobility, and long and variable Round Trip Times (RTTs).

2.1 WIRELESS NETWORK ARCHITECTURES

In the following paragraphs, a brief classification of wireless networks is provided, based on the required coverage area.

Wireless Wide Area Network (WWAN). WWANs offer connections over broad geographical areas with the use of multiple antenna sites (cells). Current WWANs are primarily based on second generation (2G) cellular technologies such as GSM and CDMA [17]. The third generation (3G) cellular networks were envisioned to replace 2G technologies, but suffered from the enormous costs for spectrum licenses as well as difficulties in identifying proper killer applications. Currently 3G technologies correspond to a smaller slice of the overall cellular market than 2G, with a high penetration evidenced in Asia Pacific and North America regions [18].



Wireless Wide-Area Network

Wireless Metropolitan-Area Network

Wireless Local-Area Network

Wireless Personal-Area Network

Fig. 1. Network Architectures.

Wireless Metropolitan Area Network (WMAN). WMANs represent a good alternative to optical fiber technologies, enabling commutations between multiple locations within a metropolitan area. The key wireless technology considered for WMANs is based on IEEE 802.16 standard [19], which is also referred as Worldwide Interoperability for Microwave Access (WiMAX). Initially, WiMAX technology was designed as a metropolitan backbone for interconnection of smaller networks or fixed individual users requiring broadband access. This is often referred to as Fixed WiMAX and corresponds to IEEE 802.16 finally approved in 2004. Then, Mobile WiMAX has been developed – an air interface modification aimed more at end-users, rather than small networks, providing the support for nomadic mobility. Mobile WiMAX is based on IEEE 802.16e standard amendment [19] approved in 2005.

Wireless Local Area Network (WLAN). WLAN technologies provide connectivity to the end-user terminal devices covering a small geographic area, like corporate or campus building. The IEEE 802.11 [20], commonly known as WiFi, became the de facto standard for WLAN networking. While the original WiFi specification approved in 1997 aimed at 1 or 2 Mb/s at the physical layer, later physical air interface modifications increased the transmission rate: 802.11a (1999) for up to 54 Mb/s in 5GHz band, 802.11b (1999) for up to 11 Mb/s in 2.4 GHz band, 802.11g (2003) of up to 54 Mb/s in 2.4 GHz band, and 802.11n for up to 250 Mb/s in both 5GHz and 2.4 GHz bands. In WiFi, mobile stations establish connections to wireless access points which serve as a bridge between the radio link and a wired backbone network. As an option, in case mobile stations are located within the transmission range of each other and no network backbone access is required, an ad hoc network may be created.

Wireless Personal Area Network (WPAN). WPANs are designed to connect user devices located within personal communication range which is typically considered of up to 10 meters from a person. Bluetooth [21] is the leading industry standard for WPANs. Nowadays, WPANs are supported by mobile phones, PDAs, laptops and other wireless devices. Nevertheless, the main application for Bluetooth remains wireless headset connection.

A promising technology in the WPAN scenario is based on Ultra-wideband (UWB) radio communications [22], potentially able to provide 1Gb/s links over short range. UWB PAN is specified in IEEE 802.15.4a standard [23] completed in March 2007.

2.2 PERFORMANCE ISSUES AND SOLUTIONS IN WIRELESS NETWORKS

Nowadays, most of the leading wireless technologies are widely deployed at the last mile – connecting end-user to the core of the network, and follow infrastructure network organization, where wireless links are mostly used to connect end user equipment to the base station which in turn provides connectivity to the fixed network.

Indeed, last mile is the most critical issue in today’s network architectures. The characteristics of the last mile links often determine the performance of the overall network, representing the actual capacity bottleneck on the entire path from the data source to the destination and influencing the characteristics of traffic patterns flowing through the network.

In addition, wireless networks suffer from several performance limitations, in some cases related to excessive burden deriving from the layering paradigm employed for the TCP/IP protocol stack design. In fact, TCP/IP originally designed for wired links (characterized by high bandwidth, low delay, low packet loss probability - high reliability, static routing, and no mobility) performs poorly in wireless domain [3, 10].

The main reasons for poor performance are in the very nature of wireless technologies and come from the advances their enable:

Mobility. One of the main advances offered by wireless networks corresponds to user terminal mobility, which allows network access from different locations while maintaining uninterrupted service. However, mobility - an essential requirement for network provisioning on anytime, anywhere basis - comes at a price.

While most of the existing wireless technologies evolve into a converged All-IP network [25], the underlying TCP/IP protocol stack reference model [26] designed for the fixed Internet does not allow smooth adaptation for mobility mainly due to its layering model [27, 28].

Traditionally, mobility management solutions resided within a single layer, with a logical division into network (layer-3) layer solutions and link (layer-2) layer solutions [29]. However, the decision about which layers should be involved in order to provide efficient mobility support represents a hot discussion topic [30, 31]. What becomes clear is that the solutions implemented at different layers are more complementary to each other rather than alternative. While some layers appear to handle mobility better than others, it is clear that mobility support cannot be implemented within a single layer in an efficient way, and thus requires cross-layer awareness and cooperation as proposed in [31].

Similar conclusion is currently driving the design of next generation cellular network followed by 3GPP group, which identifies cross-layering as the approach able to reduce handoff (handover) latency [32] fitting the requirements of many streaming, interactive, and VoIP applications.

Typical solutions aim at handoff latency reduction such as [33] and [34] and suggest notifying the network layer even before the handoff is completed at the link layer. This allows network layer to initiate and perform several handoff procedures in parallel and guarantee improved performance.

The differentiation into pre-handoff and post-handoff link layer messages is implemented by Tseng et al. in [35]. These messages are used along with cross-layer network topology information. The topology information includes logical and physical location of neighboring access points, the association between them, and location and movement direction of the mobile node, and it is used primarily to reduce probing delay and improve routing of redirected traffic.

In [36], the authors propose S-MIP (Seamless Mobile IP) architecture aimed at handling seamless handoff for Mobile IP. The main intelligence is added with the introduction of the Decision Engine (DE), which makes handoff decisions for its network domain justified on the basis of the global view of nodes connection states and their movement patterns (which allow a certain degree of prediction).

Data transfer performance in wireless networks suffers from several performance limitations such as limited capacity, high propagation delay, static routing, and high error rate. High bit error rates (BERs), which vary from 10^{-3} up to 10^{-1} for wireless links while staying between 10^{-8} to 10^{-6} for wired channels, have high impact on data transfer performance using TCP protocol, which supports the vast majority of Internet connections [37, 38]. In general, such error rates greatly degrade the performance of TCP due to the additive increase multiplicative decrease (AIMD) congestion control, which treats all losses as congestion losses and thus underestimates the actual capacity provided by the network. Moreover, the conducted research revealed that it is not always possible to compensate undesirable characteristics optimizing bit error rate at the physical, link layers or producing transport protocol adaptation to high error rates if done separately [39]. However, interlayer communication, wireless medium awareness, and joint optimization are envisioned as essential components in the field of potential solutions.

Most of the methodologies aimed at data transfer improvement in wireless networks are based on either tight interaction between the link and physical layers, or implement techniques enabling TCP layer awareness of the wireless link it operates on. For example, collided packets may not be discarded immediately, but stored in memory and then combined with future retransmissions triggered at the link layer for the purpose of joint packet decoding [40]. This technique, defined as network-assisted diversity multiple access (NDMA), exploits diversity of network resources leading to throughput performance benefits coming at the expense of increased receiver complexity.

The techniques introducing awareness of the physical medium into TCP are typically implemented using different explicit notification techniques. One of the first proposals in this category presented in [41] is Explicit Congestion Notification (ECN). It reserves a specific bit inside the IP header, which brings indication of network congestion back from a router to the sender node. This allows TCP sender to select its congestion control actions differentiating between congestion and link error related losses. The functionality of other explicit notification schemes is similar to ECN. In this way, in Explicit Bad State Notification (EBSN) [42] the sender is notified by the remote host of the bad state experienced on the link in order to reset retransmission timeouts, while in Explicit Rate Change Notification (ERCN) [43] is allowed to control TCP outgoing rate while accommodating delay-sensitive traffic needs. In [44] Sarolahti et al. proposed explicit signaling algorithm allowing network routers to increase TCP startup performance over high-speed network paths.

Having the core algorithms controlling TCP functionality such as congestion control and error recovery implemented at the sender node turns the design of optimization algorithms towards explicit notification solutions, which usually demonstrate considerable performance advantages. However, the main drawback for such solutions is the requirement for the modification of TCP sender code - traditionally implemented inside the operating system kernel, making the deployment of these schemes difficult on the wide scale. This drawback opens the possibility for receiver-side-only modifications or cross-layer schemes limiting protocol stack modifications to below IP layer - that can be implemented at the driver level or inside the interface card.

One of such schemes called ARQ proxy is presented in [45, 46]. It aims at overhead reduction deriving from the multilayer ARQ employed at the link and transport layers. To this aim, it introduces ARQ proxy at the base station and ARQ client at the mobile node agents, which substitute the transmission of the TCP ACK packet with a short link layer request sent over the radio link. As a result, ARQ proxy releases radio link resources required for TCP ACK packet transmission - which can be used by other transmitting stations.

Energy efficiency. A mobile terminal equipment relies on battery power, which imposes energy efficient operation to increase the device lifetime. Traditionally, power efficient design attempted to increase capacity of the battery and decrease the amount of energy consumed by the terminal. However, physical limitations of battery power units and high energy consumption of wireless interfaces position the main challenge of energy efficient communications into the system management domain. The main focus is devoted into joint optimization of the entire protocol stack, increasing the “sleep mode” duration for terminal transceiver – the mode with power consumption of at least an order to magnitude lower with respect to terminal transmitting or receiving modes [47], at the same moment operating a tradeoff with efficiency of network communications in terms of connectivity, data routing, scheduling, and others.

Current cellular networks include power efficient design implemented across different protocol layers. These techniques include proper modulation choice at the physical layer, channel dependant scheduling, cross-layer resource allocation schemes [48], emitting power control, smart management of idle and sleep modes – all involve tight cooperation between the mobile node and the base station, which is performed at different layers of the protocol stack. This fact positions cellular networks among the leaders of energy efficient technologies, with a typical battery life for the mobile terminal of several days.

In WiFi networks, similar optimization steps are being proposed, like Feeney et al. [47] suggest facing the problem of energy efficiency jointly at the link and network layers, or Singh et al. [49] pave the way for power-aware routing proposals by defining routing metrics - including terminal and transmission power awareness.

The problem of maintaining network connectivity with terminals spending most of the time in “sleep” mode is proposed to be solved with prediction of movements in [50], which involves cooperation between layers 1 to 3.

At the transport layer, the study shows that most of the widely used TCP modifications nowadays do not satisfy all the requirements of an energy constrained network environment [51]. This requires energy efficiency to become a part of TCP protocol design. Such a design approach is followed by Ci et al. in TCP Quick Timeout (TCP-QT) proposal [52]. TCP-QT aims at increasing the “sleep” mode duration by introducing a link layer feedback triggering fast retransmission before retransmission timeout occurrence.

While most of the above solutions require cross-layer interactions between two or at most three layers, the authors of [53] present a complete methodology for the design of cross-layer solutions aimed at energy efficient communications, including definition of the scenario, performance-cost modeling, simulation, analysis of dependency and other - which provide systematic exploration, problem partitioning and defining the cross-layer interactions that are required for optimal energy efficiency of the system.

Quality of Service (QoS). One of the first approaches for QoS provisioning, IntServ [54], was based on the idea of reservation of network resources through the entire path. The fact that IntServ required support from all network routers on the path, and it did not scale to large networks due to the requirement to maintain large number of reservations is overcome in DiffServ approach [55], which still requires support from network routers but instead of circuit-like reservation over entire network path it operates on packet basis. In DiffServ, each packet is marked with QoS requirements by the ordinator. Such requirements are typically satisfied by using multi-queue processing techniques at network routers, thus introducing prioritization. The QoS requirements are specified via Type of Service (TOS) field of the IPv4 header. However, despite of the introduction of TOS fields since IPv4 was developed [56], most of the Internet routers do not handle it. The DiffServ, being a network layer solution, provides good level of performance in wired network. However, additional medium-dependant techniques are required in the wireless domain.

The most widely considered QoS solution for wireless networks is in combination of layer-3 approaches aimed at QoS support on a network-wide scale with layer-2 approaches providing QoS at the wireless link connecting mobile users to the network core. This approach is realized in WiFi networks in IEEE 802.11e amendment [57] and WiMAX networks [19].

In [59], Firoiu et al. provide a comprehensive survey of available QoS mechanisms from technical as well as from the business perspective, demonstrating that no complete QoS solution can be obtained within a single protocol layer. In [60], the authors motivate the need for cross-layer interactions for QoS provisioning in mobile networks in order to avoid duplicating signaling efforts.

QoS provisioning and reduced energy consumption represent a challenging issue in battery equipped mobile terminals. Generally, ensuring better QoS leads to an increase in energy consumption. Solutions to this problem typically involve power-aware scheduling and QoS provisioning [61 - 64] involving tight cooperation between physical and link layers with QoS requirements specified and controlled at the application layer.

Another type of cooperation could involve transport, link and physical layers, an example of which is presented in [65]. This scheme implements dynamic resource allocation strategy synchronizing bandwidth allocation requests with TCP window evolution.

In [66], the authors propose CLA-QOS approach, which involves cooperation between application, network, and link layers of the protocol stack. The application layer at the destination monitors level of QoS constraints satisfaction in terms of packet loss ratio and feeds it back to the source node, letting the latter to adapt traffic class. The network layer is evolved into end-to-end delay measurement of network paths, which is provided to the link layer scheduler performing traffic differentiation according to the urgency factor.

Table 2.1 presents a summary of the characteristics of the main wireless network standards, aimed at underlining the common features and similarities among them and outlining the level of cross-layer design penetration. The table underlines the existence of a tradeoff between mobility, data transfer performance, energy consumption, and quality of service. One of such examples is presented in [63] where authors balance between QoS and energy efficiency. However, the general trend shows that network performance and functionalities are closely related with the level of penetration of cross-layer techniques into the design of different wireless systems.

2.3 CLASSIFYING PERFORMANCE ENHANCEMENT SOLUTIONS

Various approaches have been proposed to optimize the performance of IEEE 802.11 wireless networks. These can be broadly categorized into three groups:

- *Link Layer solutions.* The principle of this approach is to solve problems locally, with the transport layer not being made aware of the characteristics of the individual links. Such protocols attempt to hide losses in the wireless link to make it appear to be a highly reliable one. Link layer solutions require no changes in existing transport layer protocols. The proposed solutions for the link layer can be classified according to their awareness of the transport layer protocol.

TCP-unaware protocols optimize the link layer by hiding existing differences between the wireless medium and the transport layer so that the transport layer can operate as if it were installed in a wired network. This method does not violate the modularity of the protocol stack, however, since the necessary adaptations improve the reliability independent of higher-layer protocols. Nonetheless, lack of awareness can aspect performance under certain specific conditions. For instance, a link layer retransmission technique may trigger a considerable number of TCP time outs, greatly decreasing the throughput of TCP. Most of the TCP-unaware solutions, similar to the solutions presented in [1, 2], optimize link performance by compensating high error rates.

The TCP-aware link layer protocol presents certain advantages since knowledge of the protocol operating at the transport level allows fine tuning of the performance. For instance, an approach without awareness of the transport protocol may cause local link layer retransmission of a packet, as well as duplicate acknowledgement, since retransmissions can be performed on both layers. The well-knows examples of the TCP-aware protocols are Snoop [3] and WTCP [4].

Table 2.1. Characteristics of leading wireless technologies.

Technology	Mobility	Data transfer performance		Energy consumption/ battery life	Quality of Service	Cross-Layer Design Penetration
		Physical rate	Spectrum efficiency			
Wireless Personal Area Network (WPAN)						
Bluetooth (2.0)	Direct communication	Up to 2.1 Mb/s	2 bit/s/Hz	Hours	High (dedicated channels)	Low
UWB		675 Mb/s	1.35 bit/s/Hz		n/a	
Wireless Local Area Network (WLAN)						
802.11b	Nomadic subnet roaming	11 Mbps	0.55	Hours	Low (Best effort or 802.11e if employed)	Low
802.11a/g		54 Mbps	2.7			
802.11n		250 Mbps	7.22			
Wireless Metropolitan Area Network (WMAN)						
Fixed WiMAX (802.16-2004)	Fixed	10 Mb/s (max up to 70 Mb/s)	3.75 bit/s/Hz	n/a	Normal (4 traffic classes, but not supported for network wide connections)	Medium
Mobile WiMAX (802.16e-2005)	Pedestrian Mobility	2-3 Mb/s (max up to 15 Mb/s)	2 bit/s/Hz	Hours		
Wireless Wide Area Network (WWAN)						
2G (GSM)	Seamless global roaming	9.6 – 57.6 Kb/s	0.52 bit/s/Hz	Days	High (dedicated channels, voice over data priority)	High
3G (UMTS)		384 Kb/s (mobile) 2Mb/s (stationary)	Up to 2.88 bit/s/Hz			
3G LTE		100 Mb/s	5 bit/s/Hz			

- *Transport Layer solutions.* The theory underlying this approach is the modification of the transport protocol in order to achieve high throughput on wireless links. Since some packets may be lost, the modified transport protocol should implement congestion control as a reaction to packet losses, moreover, other schemes should be implemented to consider the peculiarities of the wireless environment. TCP was originally designed for wired networks, where packet losses are caused mostly to network congestion, rather than errors resulting from noisy channels, handoffs and node mobility. A reduction in the congestion window is thus the TCP reaction to packet loss of any kind. TCP modifications are logically divided into two groups according to the technique they introduce: connection splitting approach and TCP modifications.

Connection Splitting Solutions, like I-TCP [5], divide the end-to-end TCP connection is divided into fixed and wireless parts, so that more degrees of freedom are available for the optimization of the TCP over both wired and wireless links. The disadvantages of this solution mainly involve the attempt to perform transparent splitting (of the TCP) from the point of view of the TCP layer of the wired host. This leads to greater complexity in Base Station (BS) procedures, which is the most common and suitable place for splitting; the greater complexity involves not only the handling of hand-offs but also, prevention of end-to-end semantics of the TCP connection and, also greater software overhead caused by the TCP part of the stack involved at the intermediate point.

TCP modifications involve a group of solutions which promote small changes on the behavior of TCP, such as the mechanics of acknowledgement generation used by TCP. The modifications to the TCP make it unnecessary to modify the base station, thus avoiding overhead in packet delivery and the increase in BS complexity. The major proposals in this framework are summarized below.

- *Cross-Layer solutions.* Cross-layer solutions break the principles of layering by allowing interdependence and joint development of protocols involving various different layers of the protocol stack. The next section is dedicated to present cross-layer design paradigm, its advantages and possible implementations.

III. CROSS LAYER DESIGN

3.1 ISO/OSI TCP/IP PROTOCOL STACK PRINCIPLES

Currently, design of network architectures is based on the layering principle, which provides an attractive tool for designing interoperable systems for fast deployment and efficient implementation.

ISO/OSI model [16] was developed to support standardization of network architectures using the layered model. The main concepts motivating layering are the following:

- Each layer performs a subset of the required communication functions
- Each layer relies on the next lower layer to perform more primitive functions
- Each layer provides services to the next higher layer
- Changes in one layer should not require changes in other layers

Such concepts were used to define a reference protocol stack of seven layers, going from the physical layer (concerned with transmission of an unstructured stream of bits over a communication channel) up to the application layer (providing access to the OSI environment).

A protocol at a given layer is implemented by a (software, firmware, or hardware) entity, which communicates with other entities (on other networked systems) implementing the same protocol by Protocol Data Units (PDUs). A PDU is built by payload (data addressed or generated by an entity at a higher adjacent layer) and header (which contains protocol information). PDU format as well as service definition is specified by the protocol at a given level of the stack.

The same concepts are at the basis of the de-facto standard protocol stack on the Internet, namely the TCP/IP protocol stack [26].

The main advantage deriving from the layering paradigm is the modularity in protocol design, which enables interoperability and improved design of communication protocols. Moreover, a protocol within a given layer is described in terms of functionalities it offers, while implementation details and internal parameters are hidden to the remainder layers (the so-called “information-hiding” property).

3.2 The Cross-Layering Paradigm

Standardization of layered protocol stacks has enabled fast development of interoperable systems, but at the same time limited the performance of the overall architecture, due to the lack of coordination among layers. This issue is particularly relevant for wireless networks, where the very physical nature of the transmission medium introduces several performance limitations (including time-varying behavior, limited bandwidth, severe interference and propagation environments). As a consequence, the performance of higher layer protocols (e.g., TCP/IP), historically designed for wired networks, is severely limited.

To overcome such limitations, a modification of the layering paradigm has been proposed, namely, cross-layer design, or “cross-layering.” The core idea is to maintain the functionalities associated to the original layers but to allow coordination, interaction and joint optimization of protocols crossing different layers.

Several cross-layering approaches have been proposed in the literature so far [69 - 72].

In general, on the basis of available works on the topic, two approaches to cross-layering can be defined:

- Weak cross-layering: enables interaction among entities at different layers of the protocol stack; it thus represents a generalization of the adjacency interaction concept of the layering paradigm to include “non-adjacent” interactions
- Strong cross-layering: enables joint design of the algorithms implemented within any entity at any level of the protocol stack; in this case, individual features related to the different layers can be lost due to the cross-layering optimization. Potentially, strong cross-layer design may provide higher performance at the expense of narrowing the possible deployment scenarios and increasing cost and complexity.

An alternative notation is “evolutionary approach” for the “weak cross-layering” and “revolutionary approach” for the “strong cross-layering” [73].

3.3 Cross-Layer Signaling Architectures

The large variety of optimization solutions requiring information exchange between two or more layers of the protocol stack raises an important issue concerning implementation of different cross-layer solutions inside TCP/IP protocol reference model, their coexistence and interoperability, requiring the availability of a common cross-layer signaling model [74]. This model defines the implementation principles for the protocol stack entities implementing cross-layer functionalities and provides a standardized way for ease of introduction of cross-layer mechanism inside the protocol stack.

In [75], Raisinghani et al. define the goals the cross-layer signaling model should follow. They aim at rapid prototyping, portability, and efficient implementation of the cross-layer entities while maintaining minimum impact on TCP/IP modularity.

In this framework, several cross-layer signaling architectures have been proposed by the research community. While the following paragraphs will provide an overview and comparison between the most relevant solutions, it is important to note that research on the topic is far from being complete. In fact, up to now, just of few of cross-layer signaling proposals were prototyped and none of them is included into current operating systems.

A. Interlayer signaling pipe. One of the first approaches used for implementation of cross-layer signaling is revealed by Wang et al. [76] as interlayer signaling pipe, which allows propagation of signaling messages layer-to-layer along with packet data flow inside the protocol stack in bottom-up or top-down manner. An important property of this signaling method is that signaling information propagates along with the data flow inside the protocol stack and can be associated with a particular packet incoming or outgoing from the protocol stack.

Two methods are considered for encapsulation of signaling information and its propagation along the protocol stack from one layer to another: packet headers or packet structures.

- *Packet headers* can be used as interlayer message carriers. In this case, signaling information included into an optional portion of IPv6 header [77], follow packet processing path and can be accessed by any subsequent layer. One of the main shortcomings of packet headers is in the limitation of signaling to the direction of the packet flow, making it not suitable for cross-layer schemes which require instant communication with the layers located on the opposite direction. Another drawback of packet headers method is in the associated protocol stack processing overhead, which can be reduced with packet structures method.

- *Packet structures.* In this method, signaling information is inserted into a specific section of the packet structure. Whenever a packet is generated by the protocol stack or successfully received from the network interface, a corresponding packet structure is allocated. This structure includes all the packet related information such as protocol headers and application data as well as internal protocol stack information such as network interface id, socket descriptor, configuration parameters and other.

Consequently, cross-layer signaling information added to the packet structure is fully consistent with packet header signaling method but with reduced processing. Moreover, employment of packet structures does not violate existing functionality of separate layers of the protocol stack. In case the cross-layer signaling is not implemented at a certain layer, this layer simply does not fill / modify the corresponding parts of the packet structure and does not access cross-layer parameters provided by the other layers. Another advantage of packet structure method is that standardization is not required, since the implementation could vary between different solutions.

B. Direct Interlayer Communication proposed in [76] aims at improvement of interlayer signaling pipe method by introducing signaling shortcuts performed out of band. In this way, the proposed Cross-Layer Signaling Shortcuts (CLASS) approach allows non-neighboring layers of the protocol stack to exchange messages, without processing at every adjacent layer, thus allowing fast signaling information delivery to the destination layer. Along with reduced protocol stack processing overhead, CLASS messages are not related to data packets and thus the approach can be used for bidirectional signaling. Nevertheless, the absence of this association is twofold since many cross-layer optimization approaches operate on per-packet basis, i.e. delivering cross-layer information associated with a specific packet traveling inside the protocol stack.

One of the core signaling protocols considered in direct interlayer communication is Internet Control Message Protocol (ICMP) [78, 79]. Generation of ICMP messages is not constrained by a specific protocol layer and can be performed at any layer of the protocol stack. However, signaling with ICMP messages involves operation with heavy protocol headers (IP and ICMP), checksum calculation, and other procedures which increase processing overhead. This motivates a “lightweight” version of signaling protocol CLASS [76] which uses only destination layer identification, type of event, and related to the event data fields.

However, despite the advantages of direct communication between protocol layers and standardized way of signaling, ICMP-based approach is mostly limited by request-response action - while more complicated event-based signaling should be adapted. To this aim, a mechanism which uses callback functions can be employed. This mechanism allows a given protocol layer to register a specific procedure (callback function) with another protocol layer, whose execution is triggered by a specific event at that layer.

C. Central Cross-layer Plane implemented in parallel to the protocol stack is probably the most widely proposed cross-layer signaling architecture. In [80], the authors propose a shared database that can be accessed by all layers for obtaining parameters provided by other layers and providing the values of their internal parameters to other layers. This database is an example of passive Central Cross-Layer Plane design: it assists in information exchange between layers but does not implement any active control functions such as tuning internal parameters of the protocol layers.

Similar approach is presented by the authors of [81], which introduces a Central Cross-layer Plane called Cross-layer Server able to communicate with protocols at different layers by means of Clients. This interface is bidirectional, allowing Cross-layer server to perform active optimization controlling internal to the layer parameters.

Another approach, called ECLAIR, proposed by Raisinghani et al. in [75] is probably the most detailed from the implementation point of view. ECLAIR implements optimizing subsystem plane, which communicates with the protocol stack by means of cross-layer interfaces called tuning layers. Each tuning layer exports a set of API functions allowing read/write access to the internal protocol control and data structures. These API can be used by protocol optimizers which are the building blocks of the optimizing subsystem plane. This makes the optimizing system a central point for coordination of cross-layer protocol optimizers in order to avoid loops and other conflicts.

Similar goals are pursued by Chang et al. [82] with another architecture falling into Central Cross-Layer Plane category. It assumes simultaneous operation of multiple cross-layer optimization approaches located at different layers of the protocol stack and aims at coordination of shared data access, avoiding dependency loops, as well as reduction of the overhead associated with cross-layer signaling. To this aim, an Interaction Control Middleware plane is introduced to provide coordination among all the registered cross-layer optimizers implemented in different layers. The main difference of this cross-layer architecture proposal with other proposals of this category is that signaling information propagates along the protocol stack with regular data packets - making it a unique combination of Central Control Plane and interlayer signaling pipe approaches.

D. Network-wide Cross-Layer Signaling. Most of the above proposals aim at defining cross-layer signaling between different layers belonging to the protocol stack of a single node. However, several optimization proposals exist which perform cross-layer optimization based on the information obtained at different protocol layers of distributed network nodes. This corresponds to network-wide propagation of cross-layer signaling information, which adds another degree of freedom in how cross-layer signaling can be performed.

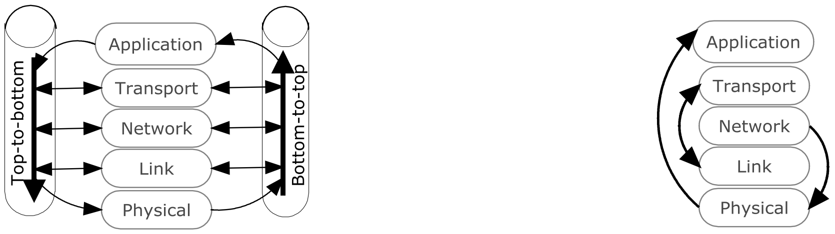
Among the methods overviewed above, packet headers and ICMP messages can be considered as good candidates. Their advantages, underlined in the single-node protocol stack scenario, become more significant for network-wide communication. For example, the way of encapsulating cross-layer signaling data into optional fields of the protocol headers almost does not produce any additional overhead and keeps an association of signaling information with a specific packet. However, this method limits propagation of signaling information to packet paths in the network. For that reason, it is desirable to combine packet headers signaling with ICMP messages, which are well suited for explicit communication between network nodes.

One of the early examples of cross-network cross-layering is the Explicit Congestion Notification (ECN) presented in [41]. It realizes in-band signaling approach by marking in-transit TCP data packet with congestion notification bit. However, due to the limitation of signaling propagation to the packet paths this notification need to propagate to the receiver first, which echoes it back in the TCP ACK packet outgoing to the sender node. This unnecessary signaling loop can be avoided with explicit ICMP packets signaling. However, it requires traffic generation capabilities form network routers and it consume bandwidth resources.

An example of adaptation of Central Cross-Layer Plane-like architecture to the cross-network cross-layer signaling is presented in [68]. The chapter suggests the use of a network service which collects parameters related the wireless channel located at the link and physical layers, and then provides them to adaptive mobile applications.

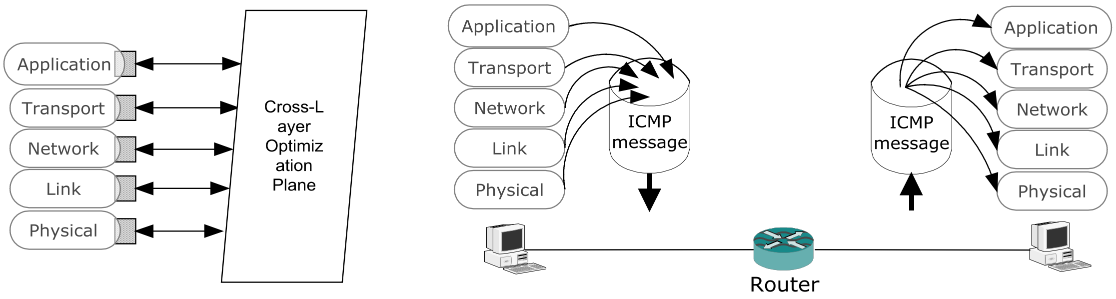
A unique combination of local and network-wide cross-layer signaling approaches called Cross-Talk is presented in [67]. CrossTalk architecture consists of two cross-layer optimization planes. One is responsible for organization of cross-layer information exchange between protocol layers of the local protocol stack and their coordination. Another plane is responsible for network-wide coordination: it aggregates cross-layer information provided by the local plane and serves as an interface for cross-layer signaling over the network. Most of the signaling is performed in-band using packet headers method, making it accessible not only at the end host but at the network routers as well. Cross-layer information received from the network is aggregated and then can be considered for optimization of local protocol stack operation based on the global network conditions.

Main problems associated to deployment of cross-layer signaling over the network, also pointed in [45], include security issues, problems with non-conformant routers, and processing efficiency. Security considerations require the design of proper protective mechanism avoiding protocol attacks attempted by non-friendly network nodes by providing incorrect cross-layer information in order to trigger certain behavior. The second problem addresses misbehavior of network routers. It is pointed out that, in 70% of the cases, IP packets with unknown options are dropped in the network or by the receiver protocol stack. Finally, the problem with processing efficiency is related to the additional costs of the routers' hardware associated with cross-layer information processing. While it is not an issue for the low-speed links, it becomes relevant for high speeds where most of the routers perform simple decrement of the TTL field in order to maintain high packet processing speed.



a) Interlayer Signaling Pipe

b) Direct Interlayer Communication



c) Central Cross-Layer Plane

d) Network-wide Cross-layer Signaling

Fig. 2. Cross-layer Signaling Architectures.

A comparison of different cross-layer signaling methods through the comparison of their essential design and deployment characteristics is presented in Table 4.1. Such features include:

- Scope defines cross-layer approach operation boundaries. Solutions which limit their operation to a single protocol stack are more flexible in the choice of signaling techniques: they can use internal protocol stack techniques such as packet structures or callback functions, thus avoiding processing related overhead and the need for standardization effort.

- Propagation latency parameter describes the delay associated with signaling message delivery. It becomes essential for signaling performed across the network, where the delay corresponds to the delay of communication links and time messages spend in router buffers. For local signaling methods, the delay is usually several orders of magnitude lower than for network-wide cross-layering. However, signaling using interlayer signaling pipe method is slower than direct interlayer communications due to layer-by-layer processing. Moreover, interlayer signaling pipe can only afford asynchronous reaction to the event occurred, while direct communication allows instantaneous reaction.

- Communication overhead parameter is more essential for network-wide communication and describes the amount of network resources needed for signaling. Encapsulation of signaling information into packets headers does not require any additional network resources in case reserved fields are used, or corresponds to just minor increase in case optional packet header fields are involved. On the contrary, ICMP messages require a dedicated effort for their delivery from the network, consuming considerable amount of network resources – including also protocol (ICMP and IP headers) overhead. The communication overhead for local signaling corresponds to the amount of operations (CPU cycles) required to deliver the message from one layer to another. This parameter is different from processing overhead, which includes message encapsulation and processing. The highest communication overhead for local communications is associated with interlayer signaling pipe due to subsequent processing at several protocol layers before message delivery.

- Processing overhead is the amount of processing power required for message creation, encapsulation, extraction, and analysis. Medium processing effort is required for signaling messages transmitted using packet headers and packet structures inside the protocol stack (mainly needed for allocation of memory and data copy procedures). Higher processing overhead is required for ICMP message creation, which involves execution of ICMP and IP layer functions of the protocol stack. For network-wide signaling, the overhead of packet headers method is medium. The procedures at the end nodes are similar to packet headers signaling performed locally, while no additional effort associated with signaling information delivery is taken. This is due to the fact that signaling information is encapsulated into the regular data packet and is being delivered along with it.

- Direction of signaling is an important characteristic which defines the applicability of the signaling approach to the chosen cross-layer optimization scheme. The schemes which do not rely on regular traffic flow (or out-of-band) signaling are packet path independent, providing a faster reaction to an event. This reaction can be preformed also in synchronous way, while packet path dependant signaling provides only asynchronous reaction. The speed and flexibility of path independent signaling comes at the expense of the additional communication resources. Nevertheless, path independence cannot be only considered as an advantage: many cross-layer optimization algorithms require signaling information associated with a specific packet transmitted through the network - making path dependant signaling more attractive in such cases. In order to implement packet association in non-path dependant approaches, a unique identification or a copy of the packet associated with the transmitted signaling information should be attached to the message. A good example of this technique is “Time Exceeded” ICMP message sent by a router for a packet dropped due to expired TTL, which includes IP header and part of data of this packet.

- Requires standardization parameter specifies whether standardization effort is needed for the cross-layer signaling method which is considered to fully support effective deployment. Standardization is required for signaling performed over the network while standardization of network protocols which are used solely inside the protocol stack of the single node is still desirable but can be avoided. This positions internal protocol signaling methods based on packet structures or callback function be less dependent on standardization bodies and thus more flexible for the deployment form the implementation point of view as well as time wise.

Table 4.1. Comparison of the Cross-Layer Signaling Methods.

Cross-Layer Signaling Method	**Scope**	**Propagation Latency**	**Communication overhead**	**Processing overhead**	**Direction of signaling**	**Requires standardization**
Interlayer Signaling Pipe						
Packet Headers	Local	Medium	High	Medium	Path dependant	√
Packet Structures	Local	Medium	High	Medium	Path dependant	×
Direct Interlayer Communications						
ICMP messages	Local	Low	Medium	High	Path independent	√
Callback functions	Local	Low	Low	Low	Path independent	×
Central Cross-layer Plane	Local	Low	Low	Low	Path independent	×
Network-wide Cross-layer Signaling						
Packet Headers	Local/Network-wide	High	Low	Medium	Path dependant	√
ICMP messages	Local/Network-wide	High	High	High	Path independent	√

The comparison among different optimization approaches should not be seen as an attempt to select the best signaling scheme to serve as an implementation basis for future cross-layer optimization solutions. On the contrary, the comparison demonstrates that no scheme is able to achieve absolute leadership in all the evaluated criteria. This fact suggests using a combination of several signaling schemes based on the specific cross-layer solution and target optimization goals.

In case of cross-layer optimization of a local protocol stack, Central Cross-Layer Plane methodology as well as Direct Interlayer Communication using callback functions seems to be appropriate, unless the optimization algorithm requires signaling of information associated with a particular packet. In such case, Interlayer Signaling Pipe using packet structures method would provide relevant performance advantages.

In case of network-wide cross-layer optimization, it is suggested to use packet headers for the transmission of periodic, delay tolerant, or information associated with particular packets. ICMP headers should be used if instant feedback is needed and size of signaling message along with associated overhead is relatively small, and avoided otherwise in order to reduce consumption of network resources.

In addition, it is necessary to evaluate the tradeoff between optimization gains and introduced signaling overhead, especially in case the proposed cross-layer optimization solution implements network-wide signaling.

3.4 CROSS-LAYER STANDARDIZATION ACTIVITIES

Nowadays, with wireless technologies driving evolution of networking, cross-layer design is envisioned as a proper solution for extending traditional TCP/IP reference model into new boundaries and overcoming its design limitations. However, this goal requires paradigm shifts by relaxing the restrictions of OSI/ISO layered model and allowing a broader view and interdependence between the layers. In [74], the authors identify the broadcast nature of the wireless link as opposed to the point-to-point nature of wired links to be the fundamental reason for limitation of the layered structure in such scenario. Furthermore, in [58], the author define cross-layering as the only option for design of Mobile Ad Hoc Networks (MANETs) along with presenting standardization proposal comprised of four parts: MANET subnetwork layer and interfaces, a heterogeneous routing protocol, hierarchical addressing schemes, and cross-layer signaling mechanisms.

While standardization process of unified framework for cross-layer design proposals or new protocol stack architectures employing cross-layer interactions is still far away, the amount of standardization proposals involving cross-layer techniques within standardization bodies is constantly increasing becoming on track in different IETF working groups. A good example of recently submitted RFC proposal to include transport-layer considerations for explicit cross-layer indications is presented in [44].

IV. CROSS-LAYER DESIGN FRAMEWORKS

The next paragraphs outline relevant ongoing activities to define proper frameworks to support cross-layer design.

4.1 MODELING CROSS-LAYER INTERACTIONS

The cross-layer approach to system design derives from enabling interaction among protocols operating at different layers of the protocol stack in order to provide improvement in terms of some performance metric.

Quantifying the effect of these interactions is very important in order to be able to systematically relate such interactions to system outcomes and be able to quantify the decision to take such interactions into account – using a cost-benefit analysis, so that the benefits outweigh the cost of additional complexity and “layer violation”. This aspect has been generally neglected in the area of wireless networks, where the discussion has been mostly qualitative or architectural, assuming the more the cross-layer interactions, the better the performance.

The subject of systematic study of cross-layering from a “system theoretic” point of view is found in the work of Law and Doyle, for example [24]. The authors follow a top-down approach to set “holistic” objectives as opposed to the wide-spread “bottom up” and “ad hoc” identification and usage of cross-layer formulations on a case by case empirical basis.

Doyle’s “NUM” methodology allows the formulation of systematic cross-layering as a “decomposition” of an optimization problem, and thus begins to address the important issues of modularity versus optimality, also other tradeoffs including signaling and locality.

A unifying analytical framework designed to bridge the gap between theory and practice is presented in [6]. It models the layers by analyzing and quantifying inter-layer interactions. This framework allows quantification of the sensitivity and optimization of the identified performance metrics with the respect to design and operating parameters across the layers. It also establishes some important guidelines for the implementation of cross-layer schemes by use of signaling architectures.

4.2 Distributed Protocol Stacks

Nowadays the networking environment significantly differs from the one the TCP/IP reference model was designed for. Current trend in networking is towards heterogeneous networking [7] with the main driving factor represented by rapid development of large scale wireless networks [8]. In this heterogeneous environment, TCP/IP shows poor performance [9, 10], driving innovation towards the identification of more cooperative cross-layer design solutions.

The concept of Distributed Protocol Stacks [11] is built up on cross-layering combined with “agent-based networking” and proxy-based design.

The Distributed Protocol Stacks architecture extends traditional layered (ISO/OSI or TCP/IP) protocol stack by allowing abstractions of “atomic” functions from a specific protocol layer and by providing means of detaching the abstracted functional blocks from the protocol stack in order to relocate them within the network (“in-network”).

Fig. 3 presents the details of the Distributed Protocol Stack architecture. The design process is composed of the following procedures: abstraction, detachment, connection, and execution.

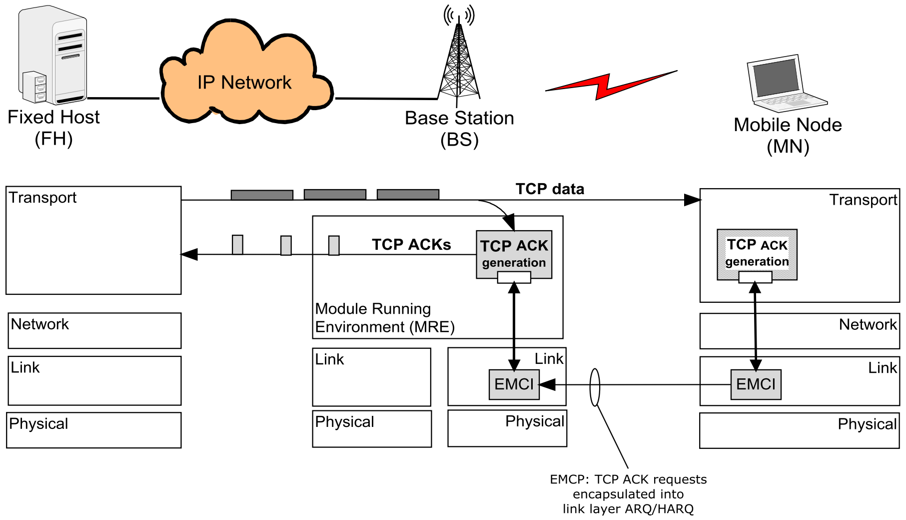


Fig. 3. Distributed protocol stack architecture.

Abstraction. Before a specific function or a set of functions of the protocol stack can be distributed over the network, they should be abstracted and detached from the protocol stack of the host node.

Identification of the functions to be abstracted depends on the optimization goal and is performed on a case-by-case basis. However, as a general recommendation an abstraction should be performed with non-time critical functions, functions which work on packet basis and do not require continuous access to the internal to the kernel structures. Ideally, abstracted functions should fit into a single functional block which operates at a packet flow basis and requires minimum or no input from the host protocol stack. The output of the abstracted functional block should be applied to the packet flow (for example controlling a single bit in a packet header), trying to avoid the requirement for direct communication with the host protocol stack.

Examples of protocol stack functional blocks that could be easily abstracted include TCP ACK generation module, header compression, IP security related functionalities, congestion related packet drop notification, advertise window adjustment in TCP, and many other.

Detachment. Once the identified set of functions is abstracted as a standalone functional block within a protocol layer, it can be detached and moved into the network. This procedure requires a certain level of cooperation from network elements (routers, switches, or gateways). In particular, network element can be considered “friendly” to the proposed Distributed Protocol Stack if they provide an environment able to support execution of the detached functional blocks – the Module Running Environment (MRE) - as an extension of their protocol stack (see Fig. 3).

MRE provides universal ways for registration and execution of different functional blocks. For example, it may provide a set of standard API functions which can be used by the host node to first transfer the abstracted functional block realized in the set of instructions understood by MRE (module description script language), and then register and run the transferred module.

Alternatively, avoiding the need for module transfer and registration procedures, functional blocks could be chosen from functional block library implemented at the network element. Execution of such blocks at the network element could be controlled by the host node or configured by network operator.

In this chapter, for sake of simplicity and aiming at providing a realistic scenario, we assume an infrastructure wireless network scenario where the mobile node detaches some of its protocol stack functions onto the base station. The base station is considered to be friendly in terms of MRE implementation, while other network nodes do not necessarily provide MRE functionality. MRE is considered to be implemented with functional blocks already installed and running at the base station - avoiding host node driven transmission, installation and configuration of the abstracted functional blocks.

Communication between the detached functional block with the host protocol stack is performed using a Module Connection Interface (MCI), which is designed to provide communication between the detached functional block and the host protocol stack.

CMI is composed of two components:

- Internal Module Connection Interface (IMCI) connects the detached functional block with MRE at the base station side, while at the host node it provides communication interface with the protocol layer the functional block has been detached.

- External Module Connection Interface (EMCI) component provides communication between the detached functional block and the host protocol stack across the network with the use of External Module Communication Protocol (EMCP).

The main idea behind CMI separation into internal and external parts is designed for the purpose of module communication overhead reduction. In particular, EMCI components could be implemented at the lower layers or the protocol stack, leading to fewer header overhead and faster processing. The communication with the IMCI located at the protocol layer where the detachment was performed is performed locally within the protocol stack and it thus does not consume network resources.

Execution of the detached functional block can be triggered by the host node using MRE module installation primitives, or can be configured and running by the base station - without requiring interaction with the host node. In the later case, the base station is responsible for notifying its clients with the information related to the list of functional blocks available. Nevertheless, it should also consider the case of operating with clients which are unaware of functional blocks running at the base station or clients which do not support such operation. Operation of the detached functional blocks should be performed in the transparent way, causing no communication performance degradation.

The design of distributed protocol stack solutions should be driven by cost / benefits analysis, where cost is related to (possibly) reduced interoperability and increased communication overhead between the detached functional block and the host protocol stack, while benefits can be measured in terms of protocol stack performance, enablers for novel user applications, or be driven from the network operator perspective.

4.3 Dynamic Adaptation and Cognitive Design

Degradation of performance due to time-varying network conditions is a challenging issue that needs to be properly addressed by current network research. The dynamic adjustment of the protocol stack parameters in a cognitive fashion is a promising approach to deal with that issue. Such idea was introduced by Mitola [12], along with the concept of cognitive radio, while the broader concept of cognitive network [13] considers system-wide goals and cross-layer design. Cognitive network is a recently emerged networking paradigm that combines cognitive algorithms, cooperative networking, and cross-layer design in order to provide real-time end-to-end optimization of complex communication systems.

Fig. 4 outlines main functional blocks of a proposed architecture enabling cooperative optimization of the protocols implemented at different layers. To some extent, the architecture proposed in [14] can be considered as a possible instantiation of the concept of cognitive network under the constraints given by interoperability with existing TCP/IP based architectures.

To operate with a standard protocol stack, each protocol layer is enhanced with a small software module able either to obtain information internal to the specific layer (observation) or to tune its internal parameters (action). The information sensed at the different protocol layers is delivered to the cognitive plane implemented at the cognitive node. This cognitive plane runs data analysis and decision making processes.

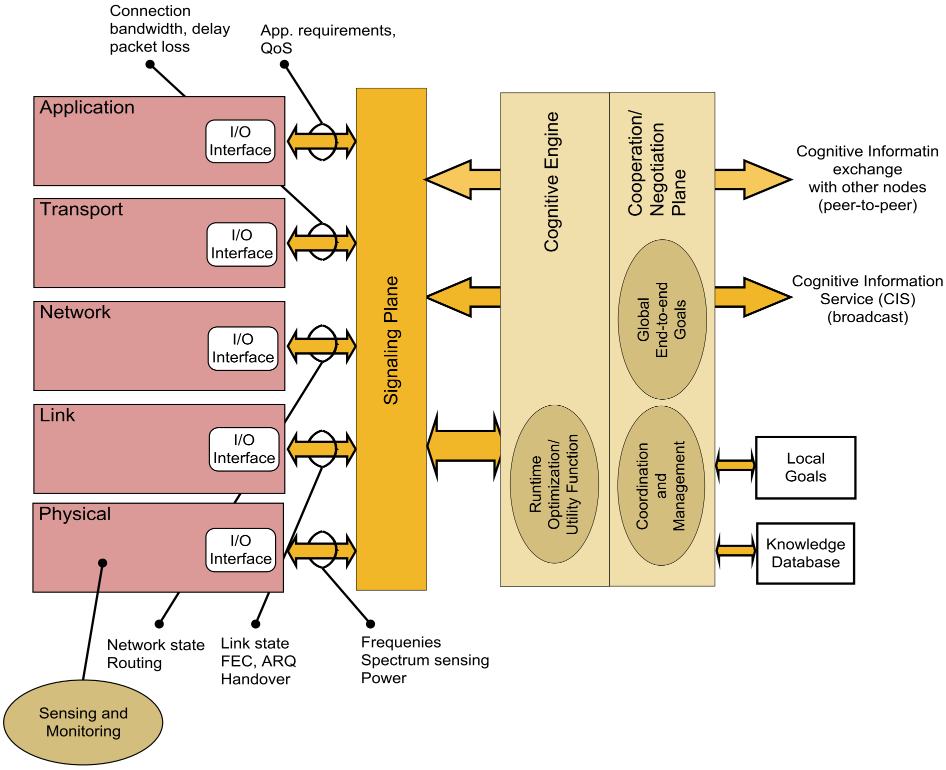


Fig. 4. Cooperative Framework.

Results of data analysis could lead to information classified as knowledge, storable in the local knowledge database.

The main task of the cognitive engine at every node is the optimization of different protocol stack parameters in order to converge to an optimal operational point given the network condition. Such operational point can be expressed by a utility function combining quality reports from all the running applications as well as other layers of the protocol stack. For that, cognitive adaptation algorithms include phases such as observation, data analysis, decision-making, and action.

The decisions made by the cognitive engine at the node aim to optimize the protocol stack performance and are driven by the goals specified in the local database. The scope of such goals is local (at node level). Most of them are generated by the demands and QoS requirements of user applications running at a given cognitive node.

Global optimization goals are defined on an end-to-end basis. Their achievement requires cooperative actions from different network nodes which are implemented using cooperation/negotiation plane operating closely with the cognitive engine.

While goals and knowledge databases are directly connected to the cognitive plane of the node and allow instant information exchange, the cognitive plane communication with the protocol stack is performed by the signaling plane. Signaling plane is responsible for providing a proper way for signaling information delivery. Depending on signaling type required, i.e. indication of parameter values, signaling threshold violation, or a callback-like indication, different signaling methods are required for parameters at different layers.

The signaling plane allows information exchange not only between the cognitive engine and different protocol layers of a single node. It also provides two interfaces for communication with other network nodes, to enable exchange of parameters' values or targeted end-to-end optimization goals. One interface operates on a peer-to-peer basis which allows information exchange between any two nodes of the network in a distributed manner. An alternate (or complementary) one, called Cognitive Information Service (CIS), corresponds to a network broadcast channel where information inserted by a given node is heard by all the nodes of the network segment. CIS signaling has obvious limitations in scalability. Because of that, it is mainly used in well-defined parts of the network with limited number of nodes, like a WiFi cell for example.

Cooperation and negotiation plane is responsible for harvesting cognitive information available at other network nodes, filtering and managing them in a distributed manner. The performed information harvesting can be performed on a scheduled basis or by using instant requests or interrupts. Moreover, information could be node-specific or parameters associated with a particular data flow.

The analysis of information gathered from cognitive nodes helps the cognitive engine to construct global knowledge and goals. Upon every adjustment, such information is reported back to cognitive nodes, so that they can adjust their appropriate local databases and, as a result, their behavior.

The main characteristic of the cognitive network architecture is scalability, assured by the use of a combination of centralized (at node level) and distributed (at network level) techniques. In particular, at node level, the core cognitive techniques (such as data analysis, decision making, and learning) are concentrated in the cognitive planes of the nodes and implemented in a centralized manner. Furthermore, observation and action software add-ons to the protocol layers serve only as instruments and cognitive planes are typically “non-intelligent”. Distributing cognitive process among the protocol layers (especially the learning and decision making functions) would require complex algorithms for synchronization and coordination between intra-layer cognitive processes. Alternatively, it seems that a single centralized cognitive process at node level brings a simpler solution, while implementation of cognitive process at network layer must be distributed or clustered implemented.

The cooperative optimization framework presented in this section aims at supporting dynamic configuration and optimization of communication protocols. It provides a way for network elements to adapt their configuration and protocol stack parameters in order to constantly adapt to changing network conditions. The process of search for optimal setup of protocol parameters is performed by using cognitive algorithms [15] and by sharing information among network nodes.

The proposed approach is based on the cooperative architecture presented in the previous section and extensively relies on quality feedback loops as well as commands allowing them to control of parameters internal to the protocol layers parameters. The core idea is to enable each node to randomly select minor variations of some parameters, test them and use the information to identify the best parameter setting given the operating context.

The main task of the cognitive plane is the adaptation of different protocol stack parameters in order to converge to an optimal operational point given the network state. This way, the cognitive adaptation algorithms include phases such as data analysis, decision-making, and action, as illustrated in Fig. 5.

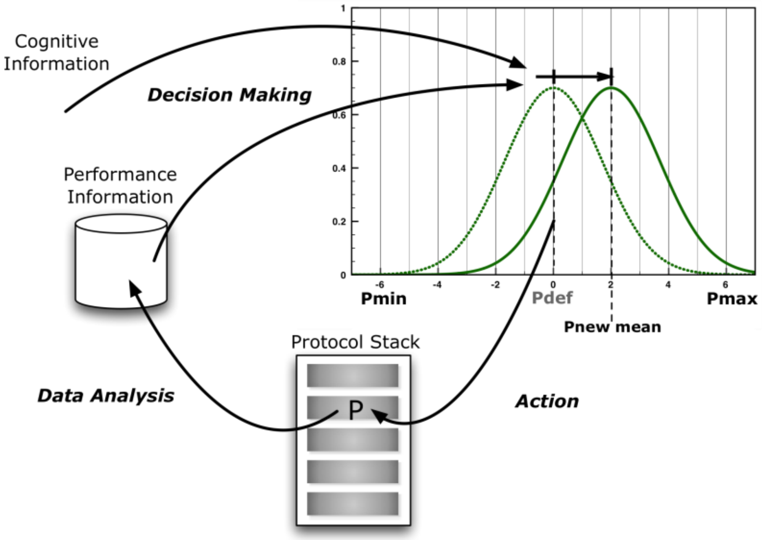


Fig. 5. Cognitive Adaptation Algorithm.

One of the main design requirements for the presented cooperative framework is to provide cognitive adaptation with minimal changes in the protocol stack. In the proposed approach, each protocol parameter P is expressed in terms of its default value P_{def} and its operation range [P_{min}, P_{max}]. The operation of the protocol is initiated with parameter P set to its default value. Then, the cognitive mechanism begins searching for optimal P values.

At the end of a given interval I, the cognitive mechanism measures using a defined quality metric and stores the obtained performance from the current value of P accordingly. Then, the mechanism selects the value of P that provides the best performance. That value is assigned to the mean of a random number generator that follows a normal distribution. Finally, a new value for P is chosen in the range [P_{min}, P_{max}] from the random number generator. The initial mean for the number generator is P_{def}. This loop continuously adjusts the mean of the normal distribution to the value of P that provides the best performance under current network conditions. The mean of the normal distribution converges to the best P value for the current network state. As a result, that value is chosen with a higher frequency. The standard deviation assigned to the normal distribution affect the aggressiveness of the mechanism in trying new values of P at each interval I.

The following paragraphs provide a numerical evaluation of the potential of cross-layering and adaptability in this framework. The described cognitive adaptation approach was implemented inside TCP congestion control algorithm. Specifically, the speed of the congestion window increase was controlled based on the observed performance using cognitive approach. Moreover two scenarios were selected based on the scope of the cognitive optimization: inter-layer and inter-node.

In inter-layer scenario cognitive information is exchanged between different layers of the protocol reference model, but no inter-node communication is implemented. Fig. 6 presents the obtained performance which demonstrates the ability of the cognitive engine to converge to the optimal value of the parameter α defining the speed of congestion window increase. For both evaluated error rates probabilities denoted as Flow F1 and F2 the cognitive approach always leads to the optimal throughput performance.

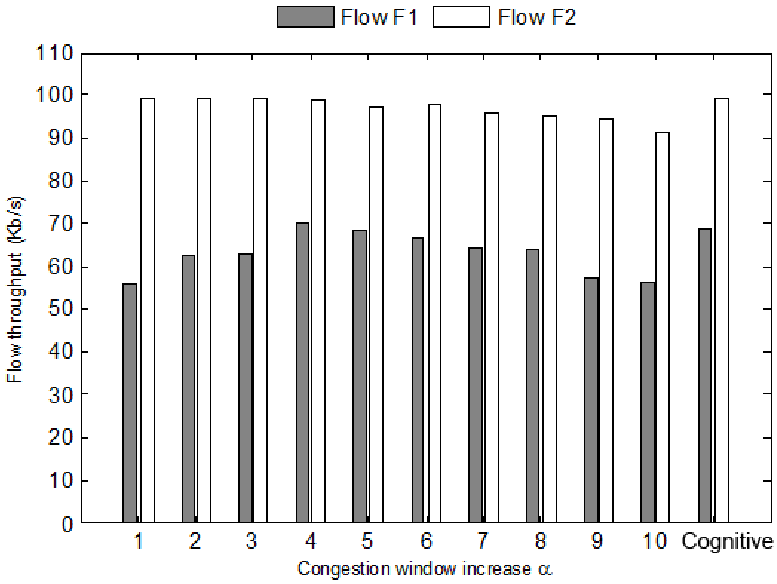


Fig. 6. Average throughput performance of a single TCP flow with different fixed congestion window increase (α) parameters and cognitive adaptation approach.

In case inter-node cognitive optimization is implemented the performance of neighboring nodes is taken into account when tuning aggressiveness of the local protocol stack. This way, possible unbalance in the assignment of network resources can be avoided.

Fig. 7 compares the performance obtained by the scenario with no cooperation involved as well as when cognitive operation is performed on inter-layer and inter-node basis for a multi-flow scenario.

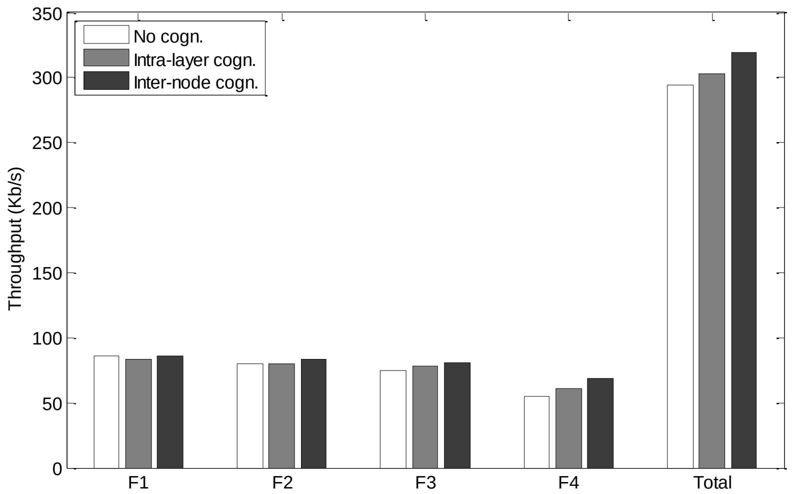


Fig. 7. Multi-flow TCP throughput performance for case with no cognitive adaptation, intra-layer cognitive adaptation, and inter-node cognitive adaptation.

As expected, the throughput decreases for long links with high error rates (F_4). The main reasons for such throughput reductions are link errors and well-known RTT unfairness for flows with different RTTs competing for the same buffer resources.

It can be observed that intra-layer cognitive engine can easily solve the problem of link losses converging to the optimal throughput value. However, it cannot cope with the problem of RTT unfairness, which requires coordination between flows. Such coordination is performed at the inter-node level.

References [83, 84] provide more details on the experiments performed.

V. CONCLUSIONS

Cross-layer design represents a suitable technology to overcome some of the current limitations of TCP/IP stack, especially in the case of wireless networks. Its core idea is to maintain the functionalities associated to the original layers but to allow coordination, interaction and joint optimization of protocols crossing different layers.

This chapter introduced the concept of cross-layering, underlining its merits and limitation, and provided information about the diffusion of such design paradigm, both in the literature as well as in existing wireless standards. Moreover, most interesting research directions were introduced and discussed.

In summary, the relevance of cross-layer design is clear in today's and tomorrow's wireless networks. However, even though thousands of contributions are available on the topic, research on cross-layering is still opening new perspectives, especially on architectural issues and dynamic adaptation of network behavior- but also on the tradeoff between performance and interoperability.

In conclusion, based on the proposed analysis of ongoing efforts, cross-layer design appears to be a suitable approach for future contributions in the framework of WLANs – able to address emerging issues related to ever-higher performance, energy consumption, mobility.

REFERENCES

[1] E. Ayangolu, S. Paul, T. LaPorta, K. Sabnani, and R. Gitlin, "AIRMAIL: A Link Layer Protocol for Wireless Networks," Wireless Networks, vol. 1, 1995, pp. 47-60.

[2] C. Parsa and J.J. Garcia-Luna-Aceves, "Improving TCP Performance over Wireless Networks at the Link Layer," Wireless Communications and Networking Conference, IEEE WCNC, p. 1253 - 1257 vol. 3, September 1999.

[3] H. Belakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance over Wireless Networks," Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (MOBICOM), Berkeley, CA, November 1995, pp. 2-11.

[4] K. Ratnam and I. Matta, "WTCP: An efficient Mechanism for Improving TCP Performance over Wireless Links," Proceedings of the Third IEEE Symposium on Computers and Communications, p. Athens, Greece, June 1998, p. 74-78.

[5] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for mobile hosts, Distributed Computing Systems," Proceedings of the 15-th International Conference, June 1995, pp. 196 - 143.

[6] F. Granelli, D. Kliazovich, J. Hui, and M. Devetsikiotis, “Performance Optimization of Single-Cell Voice over WiFi Communications Using Quantitative Cross-Layering Analysis,” 20th Iternational Teletraffic Congress (ITC'20), Ottawa, Canada, June 2007.

[7] D. Kliazovich, F. Granelli, G. Pau, and M. Gerla "APOHN: Subnetwork Layering to Improve TCP Performance over Heterogeneous Paths," IEEE Next Generation Internet Design and Engineering (NGI), Valencia, Spain, April 2006.

[8] T. S. Rappaport, A. Annamalai, R.M. Buehrer, and W.H. Tranter, “Wireless communications: past events and a future perspective,” IEEE Communications Magazine, vol. 40, no. 5, May 2002, pp. 148 – 161.

[9] C. Barakat, E. Altman, and W. Dabbous, “On TCP performance in a heterogeneous network: a survey”, IEEE Communications Magazine, vol. 38, no. 1, January 2000, pp. 40 – 46.

[10] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen, “TCP performance issues over wireless links”, IEEE Communications Magazine, vol. 39, no. 4, April 2001, pp. 52 – 58.

[11] D. Kliazovich and F. Granelli, "Distributed Protocol Stacks: A Framework for Balancing Interoperability and Optimization," International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks, Beijing, China, September 2008.

[12] J. Mitola III, “Cognitive radio for flexible mobile multimedia communications,” Mobile Networks and Applications, vol. 6, no. 5, September 2001.

[13] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, “Cognitive networks: adaptation and learning to achieve end-to-end performance objectives,” IEEE Communications Magazine, vol. 44, no. 12, pp. 51 – 57, December 2006.

[14] D. Kliazovich, N. Malheiros, Nelson L. S. da Fonseca, F. Granelli, and E. Madeira, "CogProt: A Framework for Cognitive Configuration and Optimization of Communication Protocols," IEEE International Conference on Communications (ICC), Cape Town, South Africa, May 2010.

[15] K. Machova and J. Paralic, “Basic Principles of Cognitive Algorithms Design. Proceedings of the ICCC International Conference Computational Cybernetics,” Siofok, Hungary, 2003.

[16] B. Jain and A. Agrawala, “Open Systems Interconnection,” New York, McGraw-Hill, 1993.

[17] W. C. Y. Lee, “Wireless and Cellular Communications,” 3 edition, McGraw-Hill Professional, 2005.

[18] “Worldwide Mobile Market Statistics 2007,” Portio Market Research, 2007.

[19] IEEE Std. 802.16, “IEEE Standard for Local and Metropolitan Area Networks, part 16, Air Interface for Fixed Broadband Wireless Access Systems,” IEEE Press, 2001.

[20] IEEE Std. 802.11, IEEE Standards for Information Technology -- Telecommunications and Information Exchange between Systems -- Local and Metropolitan Area Network -- Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Press, 1999.

[21] P. McDermott-Wells, “What is Bluetooth?” IEEE Potentials, vol. 23, no. 5, pp. 33- 35, 2005.

[22] R. J. Fontana, “Recent System Applications of Short-Pulse Ultra-Wideband (UWB) Technology,” (Invited Paper) IEEE Microwave Theory & Tech., vol. 52, no. 9, pp. 2087-2104, 2004.

[23] L. De Nardis and M.-G. Di Benedetto, “Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks,” 4th Workshop on Positioning, Navigation and Communication, pp. 285 – 289, 2007.

[24] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures,” Proceedings of the IEEE, vol. 95, no. 1, pp. 255 – 312, 2007.

[25] P. Newman, “In search of the all-IP mobile network,” IEEE Communications Magazine, vol. 42, no. 12, pp. s3- s8, 2004.

[26] M. W. Murhammer and E. Murphy, “TCP/IP: Tutorial and Technical Overview,” Upper Saddle River, NJ, Prentice-Hall, 1998.

[27] M. C. Chan and R. Ramjee, “TCP/IP performance over 3G wireless links with rate and delay variation,” ACM International Conference on Mobile Computing and Networking, pp. 71 – 82, 2002.

[28] S. Alfredsson, A. Brunstrom, and M. Sternad, “Transport Protocol Performance over 4G Links: Emulator Methodology and Results,” International Symposium on Wireless Communication Systems ISWCS, Valencia, Spain, 2006.

[29] T. R. Henderson, “Host Mobility for IP Networks: A Comparison,” IEEE Network, Nov. 2003, pp. 18–26, 2003.

[30] W. M. Eddy, “At What Layer Does Mobility Belong?” IEEE Communications Magazine, vol. 42, no.10, pp. 155-159, 2006.

[31] Deguang Le, Xiaoming Fu, and D. Hogref, “A review of mobility support paradigms for the internet,” IEEE Communications Surveys and Tutorials, vol. 8, no. 1, pp. 38 – 51, 2006.

[32] 3GPP, TR 25.913, “Requirements for Evolved UTRA and Evolved UTRAN,” www.3gpp.org.

[33] J. C.-S. Wu, C.-W. Cheng, N.-F. Huang, and G.-K. Ma, “Intelligent Handoff for Mobile Wireless Internet,” ACM/Kluwer Mobile Networks and Applications (MONET), vol. 6, no. 1, pp. 69 – 79, 2001.

[34] K. El Malki, “Low-Latency Handoffs in Mobile IPv4,” RFC 4881, 2007.

[35] C.-C. Tseng, L.-H. Yen, H.-H. Chang, and K.-C. Hsu, “Topology-aided cross-layer fast handoff designs for IEEE 802.11/mobile IP environments,” IEEE Communications Magazine, vol. 43, no. 12, pp. 156 – 163, 2005.

[36] R. Hsieh, Z. G. Zhou, and A. Seneviratne, “S-MIP: a seamless handoff architecture for mobile IP,” Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1774 – 1784, 2003.

[37] G. J. Miller, K. Thompson, and R. Wilder, “Wide-area Internet traffic patterns and characteristics,” IEEE Network, vol. 11, no. 6, pp. 10 – 23, 1997.

[38] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot, “Packet-level traffic measurements from the Sprint IP backbone,” IEEE Network, vol. 17, no. 6, pp. 6 – 16, 2003.

[39] F. Granelli, D. Kliazovich, and N. L. S. da Fonseca, “Performance Limitations of IEEE 802.11 Networks and Potential Enhancements,” in Yang Xiao (ed) and Yi Pan (ed), Wireless LANs and Bluetooth, Hardbound, Nova Science Publishers, 2005.

[40] M. Tsatsanis, R. Zhang, and S. Banerjee, “Network-assisted diversity for random access wireless networks,” IEEE Transactions on Signal Processing, vol. 48, no. 3, pp. 702–711, 2000.

[41] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” RFC 3168, 2001.

[42] S. Bikram, B. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, “Improving Performance of TCP over Wireless Networks,” International Conference on Distributed Computing Systems, 1997.

[43] L.-C. Wang and C.-H. Lee, “A TCP-physical cross-layer congestion control mechanism for the multirate WCDMA system using explicit rate change notification,” International Conference on Advanced Information Networking and Applications, vol. 2, pp. 449 – 452, 2005.

[44] P. Sarolahti and S. Floyd, “Cross-layer Indications for Transport Protocols,” Internet draft draft-sarolahti-tsvwg-crosslayer-00.txt, 2007.

[45] D. Kliazovich, F. Granelli, S. Redana, and N. Riato, “Cross-Layer Error Control Optimization in 3G LTE,” IEEE Global Communications Conference (GLOBECOM), Washington, DC, U.S.A., 2007.

[46] D. Kliazovich, N. Ben Halima, and F. Granelli, “Cross-Layer Error Recovery Optimization in WiFi Networks,” Tyrrhenian International Workshop on Digital Communication (TIWDC), Ischia island, Naples, Italy, 2007.

[47] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 3, pp. 1548 – 1557, 2001.

[48] H. Jiang, W. Zhuang, and X. Shen, “Cross-layer design for resource allocation in 3G wireless networks and beyond,” IEEE Communications Magazine, vol. 43, no. 12, pp. 120 - 126, 2005.

[49] S. Singh, M. Woo, and C. S. Raghavendra, “Power-aware routing in mobile ad hoc networks,” Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp. 181 – 190, 1998.

[50] Y. Dong and D. K. Y. Yau, “Adaptive Sleep Scheduling for Energy-efficient Movement-predicted Wireless Communication,” Proceedings of the 13TH IEEE International Conference on Network Protocols (ICNP), pp. 391 – 400, 2005.

[51] M. Zorzi and R. R. Rao, “Energy efficiency of TCP in a local wireless environment,” Mobile Networks and Applications, vol. 6, no. 3, pp. 265 – 278, 2001.

[52] S. Ci, S. W. Turner, and H. Sharif, “An energy-efficient TCP quick timeout scheme for wireless LANs,” IEEE International Performance, Computing, and Communications Conference, pp. 193 – 197, 2003.

[53] W. Eberle, B. Bougard, S. Pollin, and F. Catthoor, “From myth to methodology: cross-layer design for energy-efficient wireless communication,” Design Automation Conference, 2005.

[54] R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” RFC 1633, 1994.

[55] K. Kilkki, “Differentiated Services for the Internet,” Indianapolis, IN, USA, Macmillan Technical Publishing, 1999.

[56] “Internet protocol,” RFC 791, 1981.

[57] IEEE Std. 802.11e, “Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,” IEEE Press, 2005.

[58] J. A. Stine, “Cross-Layer Design of MANETs: The Only Option,” Military Communications Conference, pp. 1-7, 2006.

[59] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang, “Theories and models for Internet quality of service,” IEEE Proceedings, vol. 90, no. 9, pp. 1565 – 1591, 2002.

[60] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, “A survey of quality of service in IEEE 802.11 networks,” IEEE Wireless Communications, vol. 11, no. 4, pp. 6 – 14, 2004.

[61] J. Chen, T. Lv, and H. Zheng, “Cross-layer design for QoS wireless communications,” Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS), pp. 217 – 220, 2004.

[62] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, “Cross-Layer Design for Power Efficiency and QoS Provisioning in Multi-Hop Wireless Networks,” IEEE Transactions on Wireless Communications, vol. 5, no. 11, pp. 3306 – 3315, 2006.

[63] Z. Zhou, P. K. McKinley, and S. M. Sadjadi, “On quality-of-service and energy consumption tradeoffs in FEC-encoded audio streaming,” IEEE International Workshop on Quality of Service, pp. 161 – 170, 2004.

[64] W. L. Huang and K. B. Letaief, “Cross-Layer Scheduling and Power Control Combined With Adaptive Modulation for Wireless Ad Hoc Networks,” IEEE Transactions on Communications, vol. 55, no. 4, pp. 728 – 739, 2007.

[65] A. Narula-Tam, T. Macdonald, E. Modiano, and L. Servi, “A dynamic resource allocation strategy for satellite communications,” IEEE Military Communications Conference (MILCOM), pp. 1415 – 1421, 2004.

[66] H. Tian, S. K. Bose, C. L. Law, and W. Xiao, “CLA-QOS: A Cross-Layer QoS Provisioning Approach for Mobile Ad-hoc Networks,” TENCON, pp. 1 – 6, 2005.

[67] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet, “CrossTalk: cross-layer decision support based on global knowledge,” IEEE Communications Magazine, vol. 44, no. 1, pp. 93 – 99, 2006.

[68] B.-J. Kim, “A network service providing wireless channel information for adaptive mobile applications: I: Proposal,” IEEE International Conference on Communications (ICC), pp. 1345 – 1351, 2001.

[69] S. Toumpis and A. J. Goldsmith, “Capacity Regions for Wireless Ad Hoc Networks,” IEEE Transactions on Wireless Communications, vol. 2, no. 4, pp. 746-748, 2003.

[70] S. Pollin, B. Bougard, and G. Lenoir, “Cross-Layer Exploration of Link Adaptation in Wireless LANs with TCP Traffic,” Proc. IEEE Benelux Chapter on Communications and Vehicular Technology, 2003.

[71] L. Chen, S. H. Low, and J. C. Doyle, “Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks,” Proc. INFOCOM, 2005.

[72] X. Lin and N. B. Shroff, “The Impact of Imperfect Scheduling on Cross Layer Rate Control in Wireless Networks,” Proc. INFOCOM, 2005.

[73] F. Aune, “Cross-Layer Design Tutorial,” Norwegian University of Science and Technology, Dept. of Electronics and Telecommunications, Trondheim, Norway, 2004.

[74] V. Srivastava and M. Motani, “Cross-layer design: a survey and the road ahead,” IEEE Communications Magazine, vol. 43, no. 12, pp. 112 – 119, 2005.

[75] V. T. Raisinghani and S. Iyer, “Cross Layer Feedback Architecture for Mobile Device Protocol Stacks,” IEEE Communications Magazine, vol. 44, no. 1, pp. 85 – 92, 2006.

[76] Q. Wang and M. A. Abu-Rgheff, “Cross-layer signaling for next-generation wireless systems,” IEEE Wireless Communications and Networking (WCNC), pp. 1084 – 1089, 2003.

[77] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460, 1998.

[78] J. Postel, “Internet Control Message Protocol,” RFC 792, 1981.

[79] A. Conta and S. Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification,” RFC 2463, 1998.

[80] K. Chen, S. H. Shah, and K. Nahrstedt, “Cross-Layer Design for Data Accessibility in Mobile Ad hoc Networks,” Wireless Personal Communications, Special Issue on Multimedia Network Protocols and Enabling Radio Technologies, vo. 21, pp. 49-75, Kluwer Academic Publishers, 2002.

[81] K. M. El Defrawy, M. S. El Zarki, and M. M. Khairy, “Proposal for a cross-layer coordination framework for next generation wireless systems,” ACM International Conference On Communications And Mobile Computing, pp. 141 – 146, 2006.

[82] Z. Chang, G. N. Gaydadjiev, and S. Vassiliadis, “Infrastructure for Cross-Layer Designs Interaction,” The 16th IEEE International Conference on Computer Communications and Networks (IC3N), Honolulu, Hawaii USA, 2007.

[83] D. Kliazovich, N. Malheiros, Nelson L. S. da Fonseca, F. Granelli, and E. Madeira, "CogProt: A Framework for Cognitive Configuration and Optimization of Communication Protocols," 2nd International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT), Barcelona, Spain, May 2010.

[84] D. Kliazovich, J. Lima, Nelson L. S. da Fonseca, F. Granelli1, and E. Madeira, "Cognitive Link Layer for Wireless Local Area Networks," IEEE Latin-American Conference on Communications (LATINCOM), Medellin, Colombia, September 2009.