Proposal for a Cross-Layer Coordination Framework for Next Generation Wireless Systems

Karim M. El Defrawy
Donald Bren School of Information and Computer Science, University of California Irvine
keldefra@uci.edu

Magda S. El Zarki
Donald Bren School of Information and Computer Science, University of California Irvine
elzarki@uci.edu

Mohamed M. Khairy
Electronics and Electrical Communications Department, Faculty of Engineering, Cairo University
mkhairy@eng.cu.edu.eg

ABSTRACT
Cross-Layer design has been the focus of several recent research efforts. Due to the highly variable nature of the links used in wireless communication systems and the resource-poor nature of the wireless mobile devices, there have been multiple research efforts to improve the performance of the protocol stack by allowing cross-layer interaction in wireless systems. Cross-layer interaction means allowing communication of a layer with any other possibly non-adjacent layer in the protocol stack. Several issues related to the cross-layer design paradigm need to be addressed before it can achieve its promises. One of these issues is to have a well defined framework that manages the interaction between the different layers of the protocol stack, such that the modularity of the stack is preserved while still achieving the flexibility and adaptability which cross-layer design promises. This paper addresses this issue by proposing a cross-layer coordination framework for next generation wireless systems. The proposed framework enables the interaction between non-adjacent layers in a systematic organized way while preserving the modularity of each layer. The proposed framework addresses some of the concerns stated in recent research about cross-layer design. We believe that the existence of such a framework will ease the development of cross-layer design schemes.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network Protocols – *protocol architecture (OSI model).*

General Terms
Algorithms, Design.

Keywords
Cross-Layer Design, Signaling Framework, Cross-Layer Adaptation, Video over Wireless.

1. INTRODUCTION
The rapid growth of wireless technologies has highlighted some shortcomings in the current protocol stack. This stack served well for the wire-line environment but has several limitations when used as is in the wireless environment. For example some transport protocols suffer severely from the consequences of high bit error rates. The widely used TCP protocol is the primary example of this situation [2]. Cross-Layer design has emerged as a solution to these limitations. Due to its promises to achieve performance improvements in future networks it has attracted a lot of attention. Several issues concerning the cross-layer design paradigm need to be addressed before it can achieve its promises and be regarded as a mature solution. This paper presents an overview of recent research in cross-layer design. It identifies an issue that has to be solved before cross-layer design can be considered a reality, namely the issue of defining how the non-adjacent layers communicate with each other. The rest of the paper is organized as follows: in section 2 we summarize the recent research activities in cross-layer design. In section 3 we explain the details of our proposed cross-layer coordination framework. Section 4 presents an example of an application that uses this framework to improve performance, namely the transmission of video over wireless channels using cross-layer design. We conclude and highlight the future work in section 5.

2. RESEARCH IN CROSS-LAYER DESIGN
Several papers discuss the problems that could be solved via cross-layer coordination between the different layers of the protocol stack. The authors in [8] identify and discuss the importance of cross-layer design in the future of wireless communication systems. They explore key issues which are likely to impact cross-layer network design. They also consider the gains that could be achieved by means of a cross-layer approach for wireless network design, where physical layer information is passed to the higher layers, and discuss some of the standardization efforts that are beginning to move towards this integrated cross-layer approach.

The TCP/IP protocol stack which is widely used today is shown in Figure 1, in this stack the traditional behavior is that communication occurs only between adjacent layers. In order to facilitate a cross-layer behavior the concept of coordination planes was introduced in [2]. A coordination plane is a cross-section view of the protocol stack on which interlayer coordination algorithms can be applied, and it is focused on solving a set of problems of the same kind. Four coordination planes were discussed in [2], namely:

1. Security: The main purpose of this coordination plane is to eliminate multiple layers of encryption, which show up frequently in communication systems.

2. Quality of Service: The QoS plane is responsible for distributing the QoS requirements and restrictions along the whole protocol stack, and coordinating the efforts across the layers.

3. Mobility: This plane deals with the problems created by mobility scenarios, such as the poor interactions between TCP congestion-control, Mobile IP, and layer two mobile solutions, as well as mobility across networks.

4. *Wireless link adaptation:* This plane is used to represent the cross-layer issues related to a given wireless link layer, and includes bit error rate and bit rate adaptability. This plane excludes mobility.

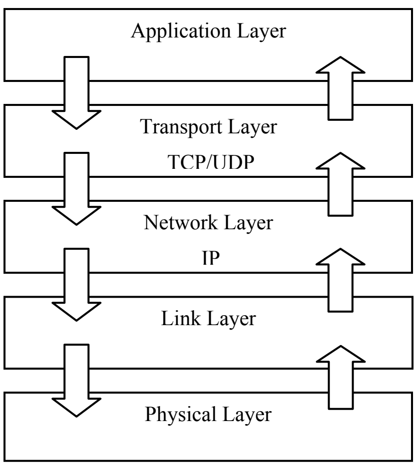

Figure 1. TCP/IP protocol stack.

In [7], different cross-layer signaling schemes are discussed and compared qualitatively. A cross-layer signaling scheme called Cross-Layer Signaling Shortcuts (CLASS) is proposed. The authors present a set of evaluation criteria and compare CLASS with current cross-layer signaling schemes. Possible application areas of CLASS are identified, and a reference application program is presented for applying CLASS into various cross-layer management areas.

In [3] key design issues associated with providing multimedia service over wireless networks are presented. The authors of [3] focus on enhancing the transmission of video over wireless through the adaptation to a wireless environment through a cross-layer design framework, which promotes communication and interaction across multiple protocol layers.

Some papers focus on providing solutions to specific problems e.g. power management, QoS …etc; these solutions are proposed as adaptation algorithms that require the communication and coordination between different layers of the protocol stack. The authors in [4] propose a joint cross-layer design for QoS content delivery. They propose a cross-layer scheme among application, MAC and physical layer, for wireless QoS content delivery. Their proposed QoS-aware scheduler and power adaptation scheme at both uplink and downlink MAC layer controls the behavior of the physical layer for resource efficiency. In [6] the authors propose a scheme where the MAC layer adapts its behavior to transport and application layer requirements as part of a new framework to support QoS in wireless LANs. They utilize cross-layer interaction between layer 2, 4 and 7 (i.e. link, transport and application layers) to achieve MAC-layer adaptation.

In [5] the authors define the main problems that might meet cross-layer design in general and offer basic design principles which are summarized in four main points:

1- Interactions and the Law of Unintended Consequences: The designer must consider the effect of a particular cross-layer interaction between certain layers on another seemingly unrelated, part of the protocol stack.

2- Dependency Graph: Where the designer represents protocols graphically as interactions between parameters. In this graph every relevant parameter is a node, and a directed edge indicates the dependency relation between the parameters. By combining the graphs for various protocols, a dependency graph for the entire stack can be obtained. This graph can later be used to prevent adaptation loops.

3- Time-Scale Separation and Stability: From the previously constructed dependency graph some stability principles could be derived.

4- The Chaos of Unbridled Cross Layer Design: Other design issues include whether if several cross-layer interactions are implemented, will they affect the performance as well as the cost of such addition of interactions.

The authors in [9] present an overview of up-to-date cross-layer proposals and develop a taxonomy to classify them. [9] also summarizes the solutions for implementing cross-layer interactions available in the literature. [9] concludes with listing open challenges in the field of cross-layer design such as identifying the important cross-layer couplings, the coexistence of cross-layer design proposals and the standardization of interfaces.

We propose a framework which organizes how the layers communicate, coordinate and exchange information with each other. This framework makes it possible to incorporate an algorithm that makes use of cross-layer interaction to solve a certain problem without requiring major changes to the protocol stack. We believe that our proposed solution which we call the "Cross-Layer Coordination Framework" addresses some of the concerns of the authors of [5] and [9].

3. PROPOSED CROSS-LAYER FRAMEWORK

3.1 Requirement for the Framework

While designing the cross-layer framework, it was of high importance to fulfill some requirements in order to achieve an efficient, scalable and modular design. These requirements were:

1- Modularity: It is desirable to preserve the modularity of the protocol stack layers while enabling interaction between them in an organized way.

2- Scalability: It should be possible to easily add several new adaptation algorithms to a particular layer or different layers without affecting the performance or other parts of the protocol stack.

3- *Efficiency:* The cross-layer interactions should not slow down or degrade the performance of the protocol stack.

3.2 Proposed Framework Overview

From the above mentioned criteria and the published literature discussing the cross-layer architecture, the following proposed design achieves the required goals while being realistic, practical and easy to implement.

In [7], different cross-layer signaling schemes are discussed and compared qualitatively. One scheme utilizes a network service server which stores all the cross-layer information and the communication between different layers happens through this intermediate server. This approach has its benefits but is too slow due to the existence of the server on a remote host.

Using the same concept but locating the cross-layer coordination server in the local host will achieve the desired cross-layer coordination functionality while eliminating the overhead of remote communication. Cross-layer coordination clients are to be added to each layer of the protocol stack to enable the interaction with the server. An overview of the proposed cross-layer framework is demonstrated in Figure 2.

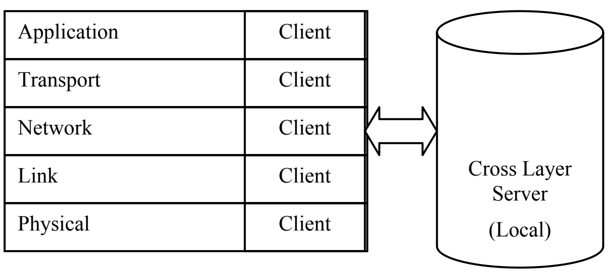

Figure 2. Overview of cross-layer coordination framework.

A simple operational scenario of the framework would be that the client in a certain layer sends an event to the cross-layer server. This event is forwarded to the target layer (or layers) by the server. The client of the target layer receives the event from the cross-layer server and performs the necessary adaptation algorithm.

Events are bi-directional notifications from client to the server and vice versa. An example of an event would be "fading_start" which is sent from the physical layer to other layers through the server when a long fading situation is detected in the wireless channel. When the server receives a certain event it triggers or "wakes up" the server-side management algorithm. The management algorithm is responsible for taking the appropriate action based on the event type and the state variables of different layers. This action could be forwarding the event to other layers or just updating a parameter in the server.

3.3 The Cross-Layer Framework's Main Building Blocks

In order to fully address all aspects of the framework, the following issues have to be discussed in detail.

- Cross-layer Client (to be added to each layer of the traditional protocol stack to enable the cross-layer coordination operation)
- Cross-Layer Server
- Signaling Scheme (Event Messages)
- Adaptation Algorithms (reside in the cross-layer client)

Breaking the cross-layer framework into these four main blocks makes it easier to handle each required aspect of the framework independently and focus on its solution.

3.4 Details of the Cross-Layer Coordination Framework

This section presents a detailed description of each part of the proposed cross-layer framework. The details of the cross-layer framework are shown in Figure 3.

3.4.1 The Cross-Layer Client

The cross-layer client is the part which is added to each layer of the protocol stack to facilitate the cross-layer coordination functionality. The client communicates with other clients in other layers through the cross-layer server to achieve the required functionality. The cross-layer client consists of two major parts, namely:

3.4.1.1 The Adaptation Module

The adaptation module could be divided into three main parts:

1. The adaptation algorithm itself. This is the logic and the implementation which solves a certain problem. It receives events from the server and sends events to it and can change the state variables of the layer it resides in. It communicates events to other layers through the server.
2. The conversion of the received parameters from other layers into the form that the algorithm needs to operate on.
3. The parameters from other layers which might be necessary for the proper operation of the algorithm.

3.4.1.2 Abstracted Layer State

Each layer in the protocol stack could be viewed as a set of parameters. Depending on the value of these parameters, one can determine the overall state of the layer and determine its behavior (i.e. each layer could be abstracted in a set of parameters).

An example of that would be:

- For a TCP connection (Transport Layer): Window Size, RTT …etc.

- For a PHY connection: Modulation Order, Code Rate, SNR …etc.

3.4.2 The Cross-Layer Server

The cross-layer server resides outside the protocol stack to facilitate the cross-layer coordination functionality. It could be viewed as a service or part of the operating system. The cross-layer server consists of two major modules, namely:

3.4.2.1 Control Module

The Control Module is divided into two parts:

1. The "Action Module" which takes the actions towards other layers, i.e. sends events to them in the form of

"Event Messages", or takes internal actions in response to a certain event received from a client.

2. The "Event Management Module" which manages concurrent events and schedules which event to be handled first in the case of several events occurring at the same time. This form of scheduling should be easy and not time consuming; otherwise it will affect the performance of the system.

3.4.2.2 Parameter Management Module

This Module consists of the "Parameter Repository" which is responsible for saving the parameters (abstracted state) of each layer in a suitable form for the other layers to access easily.

3.4.3 Signaling Scheme (Event Messages)

As mentioned previously, the communication between different non-adjacent layers in the protocol stack happens through the cross-layer server. When an initiating layer wants to send a certain event to another target layer, the client of the initiating layer sends this event to the server, which forwards it to the target layer. The event message should be expressive enough to carry the necessary information from one layer to another. Different events could achieve different tasks, for example an event could be used to inform the server of a change of a parameter and report its new value. Another event could be used to request a certain parameter value from the server. The server could send events to the clients to request a certain action to be performed or request from the client a certain parameter value. An event is transmitted from a client to the server via an "Event Message". An "Event Message" could contain one event or several events with or without associated parameters. Each event should have a priority to facilitate scheduling it among other events. Event parameters are optional (i.e. there could be an event without a parameter). Events should be encoded in a TLV (Type, Length, and Value) to facilitate the existence of several events per message. A proposed "Event Message" structure is shown in Figure 4. Each of the fields shown in the message should be encoded in the TLV format.

3.5 What the Cross-Layer Designer needs to specify

In order to utilize the proposed framework, the cross-layer protocol designer needs to specify the following parts in the cross-layer framework:

1. The adaptation algorithm which will be implemented inside the client of a certain layer or several layers of the protocol stack.

2. The necessary events, their numbers, types, their parameters and the associated action with each event, which will be sent from a client to the server and vice versa.

3. A priority policy for differentiating between different events in case they are all present at the same time at the server side. This policy should define which one to execute first when several events of different types are present.

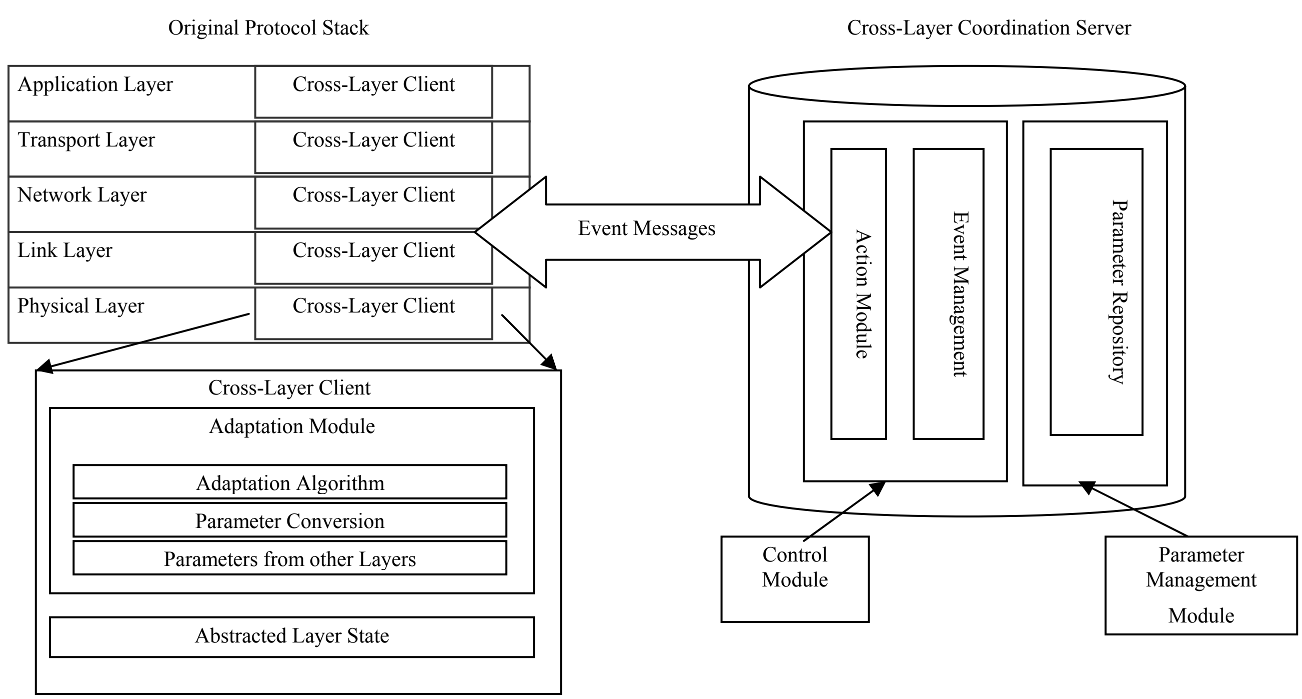

Figure 3. Details of cross-layer coordination framework.

4. VIDEO TRANSMISSION USING CROSS-LAYER DESIGN

This section illustrates how the framework is used in the case of the transmission of video over a wireless channel using cross-layer operation.

The authors in [1] propose a novel transmission scheme for real time video over wireless which utilizes cross-layer interaction between the physical layer and the video application layer. In their proposal, the physical layer determines the number of bits that it can transmit each coherence period. This information is supplied to the video application which adapts its transmission based on that number to achieve the best performance.

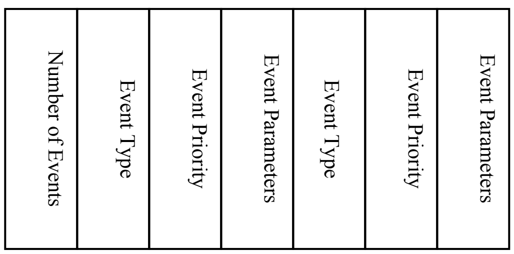

Figure 4. Event message structure.

The technique makes use of combined Progressive Group of Pictures (PGOP) and Fine Grained Scalable (FGS) video to adapt the video transmission to the wireless channel.

The FGS video is a two layered scheme comprising a base layer (BL) and an enhancement layer (EL). With the PGOP scheme, the BL rate can be maintained close to a predetermined constant. The BL guarantees an acceptable video quality. The EL can be received even partially. The reception of additional EL bits can only increase the video quality. FGS is used in the proposal due to its flexibility of arbitrary bit rate truncation at the EL [1]. This is utilized as an advantage to adapt to the fluctuating channel capacity.

In the proposed cross-layer solution for rate control, the encoder does not vary the transmission rate every coherence period. The encoder provides all the frame bits in a buffer in a continuous fashion for the packetization process to truncate. The packetization process will choose the maximal number of bits to transmit based on the feedback from the physical layer in each coherence period and will truncate the remaining bits. The packetizer does not have to know the semantics of the data that needs to be truncated as arbitrary truncation of FGS video is possible. The packetizer selects only the necessary bits from the beginning of the buffer and truncates the rest. The packetizer chooses only the transportable number of bits minus the lower layer packet overhead (headers of lower layers) of the layers below it. The total number of bits available at the physical layer for transport after including packetization would be the exact number of transportable bits for that coherence period. This scheme provides real-time rate adaptation for every coherence period.

We use this application as an example of how our cross-layer framework could be used to allow cross-layer coordination. The requirement in this case is that the client of the physical layer needs to inform the client of the video application the number of bits which it can transmit during each coherence period. This corresponds to an event that the physical layer needs to send to the video application via the cross-layer framework.

This event is transported in an "Event Message" which would be in the form shown in Figure 5. The message carries one event which is the number of bits that the channel state allows to be transmitted during the next coherence period denoted by X. The event priority is high, since if the video application does not receive this number it will not be able to construct the video packet in the correct form. There could be a default value which is the minimum value which corresponds to only accommodate the BL bits.

The cross-layer server upon receiving this message from the physical layer client determines its event type. The management module in the server then decides to forward the event to the video application. Thus another message from the server to the client of the video application is needed. This message has the same type and format as the one between the physical layer client and the cross-layer server.

The client in the video application receives this event and starts performing the adaptation algorithm which is shown in Figure 6. The operation of the algorithm is very simple, the algorithm drops a number of bits from the buffer filled by the video encoder so that the remaining number of bits is equal to the number supplied by the physical layer minus the overhead of lower layer protocols (RTP, UDP, IP and Link layer headers).

Note that the demonstration of the adaptation algorithm does not take into account the occurrence of an error that prevents the event from reaching the video application. Several strategies could be used to solve this issue, for example sending the BL bits only or sending the same number of bits as the last packet, or having a weighted average of the sizes of several previously transmitted packets.

In this case no policy is required for determining how this event should be handled in the case of the existence of other events at the same time. An example of such a policy would be if the physical layer sends the event "fading_start", which indicates that a sudden fade has occurred in the wireless channel, meaning that the channel is inaccessible, in this case it should be forwarded to the video application which should in turn drop the packet it was constructing.

The abstracted layer state of the video application contains the encoding rate that the video is using and the number of bits in the BL and EL and any additional information for example the encoding algorithm used.

The parameter repository in the cross-layer server in this case will be used to store the base rate of the BL at which the video application is sending and the maximum rate corresponding to BL + EL. It also is used to store the values that the physical layer reported for the previous coherence periods.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented an overview of recent research in the area of cross-layer design. We identified an essential issue that we believe is crucial to the success of the cross-layer design scheme. This issue is the existence of an organized framework which defines how the non-adjacent layers communicate. We proposed a solution to this issue in the form of a general framework which achieves the desired functionality. The details of the framework were discussed and presented. Finally, the benefits of the proposed framework were highlighted and the role of the cross-layer protocol designer was explained. We concluded by giving an example demonstrating how to use the proposed framework to implement a novel adaptive cross-layer video transmission technique. We are currently implementing a prototype of our proposed framework to test it in real operational scenarios. Future work includes investigating the impact of implementing several other cross-layer adaptation algorithms under the same framework and qualitatively assessing the performance of the framework and applications under such operation.

Number of Events = 1	Event Type = Number of Bits to Transmit	Event Priority = High	Event Parameters = X

Figure 5. Event message for cross-layer video example.

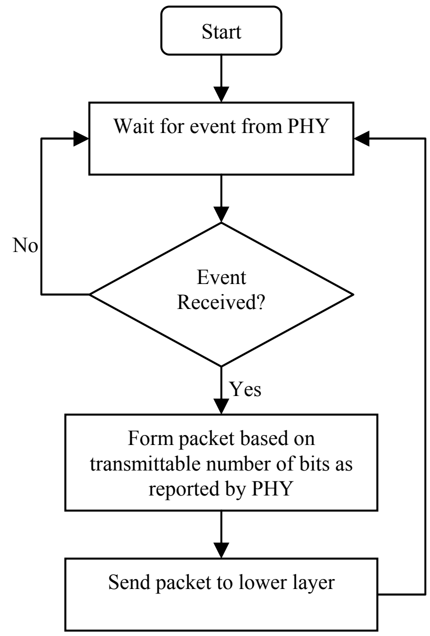

Figure 6. Adaptation algorithm for video application.

6. REFERENCES

[1] Arulsaravana Jeyaraj, Liang Cheng, Magda El Zarki. Proposal for a Cross Layer scheme for Real Time Wireless Video. In preparation, April, 2006.

[2] Carneiro, G.; Ruela, J.; Ricardo, M. Cross-Layer Design in 4G Wireless Terminals. IEEE Wireless Communications Magazine. Volume 11, Issue 2, Apr. 2004 Page(s):7 – 13.

[3] Haitao Zheng. Optimizing wireless multimedia transmissions through cross layer design. IEEE Proceedings of International Conference on Multimedia and Expo, 2003 (ICME '03). Volume 1, 6-9 July 2003 Page(s):I - 185-8 Vol.1.

[4] Jie Chen; Lv, T.; Haitao Zheng. Joint cross-layer design for wireless QoS content delivery. IEEE International Conference on Communications, 2004. Volume 7, 20-24 June 2004 Page(s):4243 - 4247 Vol.7.

[5] Kawadia, V.; Kumar, P.R. A cautionary perspective on cross-layer design. IEEE Wireless Communications Magazine. Volume 12, Issue 1, Feb. 2005 Page(s):3 – 11.

[6] Pau, G.; Maniezzo, D.; Das, S.; Yujin Lim; Pyon, J.; Heeyeol Yu; Gerla, M. A cross-layer framework for wireless LAN QoS support. Proceedings of International Conference on Information Technology: Research and Education (ITRE2003). 11-13 Aug. 2003, Page(s):331 – 334.

[7] Qi Wang; Abu-Rgheff, M.A. Cross-layer signalling for next-generation wireless systems. IEEE Wireless Communications and Networking conference, WCNC 2003.Volume 2, 16-20 March 2003 Page(s):1084 - 1089 Vol.2.

[8] Sanjay Shakkottai, Theodore S. Rappaport and Peter C. Karlsson. Cross-Layer Design for Wireless Networks. IEEE Communications Magazine. Volume 41, Issue 10, Oct 2003 Page(s):74 – 80.

[9] Srivastava, V.; Motani, M. Cross-layer design: a survey and the road ahead. IEEE Communications Magazine. Volume 43, Issue 12, Dec. 2005 Page(s):112 – 119.